WiDeo: Fine-grained Device-free Motion Tracing using RF Backscatter

Kiran Joshi, Dinesh Bharadia, Manikanta Kotaru, and Sachin Katti, *Stanford University*

https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/joshi

This paper is included in the Proceedings of the
12th USENIX Symposium on Networked Systems
Design and Implementation (NSDI '15).

May 4–6, 2015 • Oakland, CA, USA

ISBN 978-1-931971-218

Open Access to the Proceedings of the
12th USENIX Symposium on
Networked Systems Design and
Implementation (NSDI '15)
is sponsored by USENIX

WiDeo: Fine-grained Device-free Motion Tracing using RF Backscatter

Kiran Joshi, Dinesh Bharadia , Manikanta Kotaru, Sachin Katti
krjoshi, dineshb, mkotaru, skatti@stanford.edu

Abstract

Could we build a motion tracing camera using wireless communication signals as the light source? This paper shows we can, we present the design and implementation of WiDeo, a novel system that enables accurate, high resolution, device free human motion tracing in indoor environments using WiFi signals and compact WiFi radios. The insight behind WiDeo is to mine the backscatter reflections from the environment that WiFi transmissions naturally produce to trace where reflecting objects are located and how they are moving. We invent novel backscatter measurement techniques that work in spite of the low bandwidth and dynamic range of WiFi radios, new algorithms that separate out the moving backscatter from the clutter that static reflectors produce and then trace the original motion that produced the backscatter in spite of the fact that it could have undergone multiple reflections. We prototype WiDeo using off-the-shelf software radios and show that it accurately traces motion even when there are multiple independent human motions occurring concurrently (up to 5) with a median error in the traced path of less than 7cm.

1 Introduction

Fine-grained human motion tracing, i.e. the ability to trace the trajectory of a moving human hand or leg or even the whole body, is a general capability that is useful in a wide variety of applications. For example, it can be used for gesture recognition and virtual touchscreens (e.g. Kinect style natural user interfaces), activity recognition (e.g. controlling the Nest thermostat depending on intensity of human activity), monitoring of young infants and the elderly, or security applications such as intruder detection. Motivated by these applications, the computer vision community has developed a number of depth sensing based systems (e.g Kinect) to implement motion tracing capabilities in cameras. However these devices are limited because they have a constrained field of view (around 2-4m range with a 60 degree aperture), and do not work in non line-of-sight scenarios, preventing their use in many applications such as whole home activity recognition, security and elderly care.

To tackle these limitations, recent work namely RF-IDraw [43] - has built a motion tracing system using wireless signals. The idea is that users would wear RFID tags, and the motion tracing system would generate transmissions and then listen to reflections of wireless signals from these tags. RF-IDraw then infers the underlying hand motion from changes in reflection signal parameters such as angle of arrival over time. RF-IDraw demonstrates good accuracy and since it uses lower frequencies than light (the 900MHz RFID band whereas visible light is at 600THz), it works in non line-of-sight (NLOS) scenarios and in the dark. However, RF-IDraw has two limitations that restrict its deployability. First, RF-IDraw requires the user whose motion is being traced to wear a special RFID tag on her hands. However, users are accustomed to motion tracing using systems such as Kinect that do not require the user to have any special hardware on them, and changing user habits can be hard. Second, the tracing system requires large antenna arrays of eight antennas with a separation of 8λ, that in their current implementations translates to an antenna array distance of nearly 2.62m. Expecting users in homes to deploy antenna arrays that might span almost an entire room is a big hurdle.

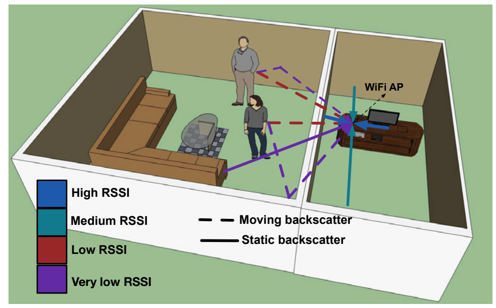

Figure 1: *WiDeo in operation: The compact WiFi AP in the study integrates WiDeo's motion tracing functionality, and can reconstruct the hand movement made by humans in the living room. WiDeo traces motion even though the AP is separated by a wall and does not have a LOS path to the humans, and doesn't require that the humans have any RF devices on them.*

Fig. 1 depicts our goal which is to design a device free, compact motion tracing system. By device free we mean that the humans whose motion is being traced do not need to have any devices on them, whether it's RFID tags or phones. By compact we mean that the motion tracing is implemented on standard WiFi or LTE APs (albeit with minor modifications in hardware and software) and the APs have antenna arrays that they would have had as standard APs anyways. Thus the system is as compact as an AP that is already being deployed. Finally, we would like the system to be non-intrusive, it should be integrated into WiFi and LTE APs that people anyway deploy in their homes and reuse existing packet transmissions for fine-grained motion tracing.

The above requirements pose unique challenges. First, since the system needs to be device-free, it can only rely on natural reflections of the transmitted signals that human limbs naturally produce. These are relatively weak compared to the ones from RFID tags that RF-IDraw uses, and reflections from different objects in the environment cannot be easily distinguished since they are all slightly distorted copies of the same transmitted signal (each RFID tag has its own unique IDs which allows RF-IDraw to distinguish different moving hands because the tags will be different). Second the fact that the system uses a compact antenna array with at most four antennas and regular spacing of $\lambda/2$ makes achieving high spatial accuracy difficult. As the RF-IDraw paper notes, regularly spaced, compact antenna arrays struggle to resolve the spatial angles of incoming signal reflections.

We present WiDeo, a device-free, compact motion tracing system with standard AP antenna arrays. WiDeo only needs 4 antennas per AP, with a spacing of $\lambda/2$ which translates to an antenna array length of 18cm for WiDeo-integrated WiFi APs. At a high level, WiDeo uses the AP's transmitted signals itself as a flash to light up the scene, and then analyzes the *natural reflections* of these transmitted WiFi communication signals from the environment that arrive back at the AP over time to trace any motion that's occurring. WiDeo accomplishes motion tracing through three main components which operate in sequence:

Backscatter Sensor: The sensor analyzes the composite reflected signal received at the WiDeo AP (referred as backscatter) to tease apart the individual reflections coming from each significant reflector in the environment, and calculates each reflection's amplitude, time of flight (ToF) and angle of arrival (AoA). Our key contribution here is a novel algorithm that accurately estimates these backscatter components in spite of the constraints that the humans are device-free, and the limited spatial resolution of the compact antenna arrays. Our key insight is to exploit the natural sparsity that exists in indoor environments; as several empirical studies on indoor MIMO [16, 19] have shown, the number of significant reflectors in an environment is fairly small. WiDeo exploits this insight to accurately measure the backscatter parameters using sparsity aware optimization algorithm.

Second, WiDeo must tolerate limited dynamic range, which causes strong reflections to swamp weak ones, and limited sampling bandwidth, which hides reflections spaced closely in time. Typical WiFi sampling of 80Msps implies a resolution of 12.5ns, or about 6 feet. Our novel algorithms separate weak and closely-spaced reflections despite the limitations of commodity radios.

Declutterer: Reflectors abound in indoor environments, and most of them will be static. The declutterer analyzes the raw set of reflection parameters estimated by the backscatter sensor, and clusters them into groups that correspond to reflections from static and moving reflectors. Further it also eliminates the static reflectors since they are not useful for motion tracing and enables WiDeo to specifically focus on reflections arising from moving objects.

Motion Tracing: This component of WiDeo analyzes the reflections arising from moving objects to predict the underlying motion that could have produced those sequence of reflections and their parameters. We design a novel statistical and sequential estimation framework that predicts the motion that might have taken place, then estimates the changes in reflection parameters the predicted motion would have produced, and compares it with the actual estimated reflection parameters from the backscatter sensor to continuously refine WiDeo's estimate of the motion that occurred.

We design and implement a prototype of WiDeo using WARP radios and simulation environment. The radios are running a standard WiFi OFDM PHY using up to 40MHz, and use 4 antennas with a spacing of 6cm for an overall length of 18cm. We conduct experiments in indoor environments to demonstrate the accuracy of WiDeo's motion tracing. We show that WiDeo can accurately trace multiple sets of fine-grained motion with a median tracing error of less than 7cm, which is comparable to RF-IDraw's performance of tracing error of 5.5cm. Further, the motion tracing has very high resolution, WiDeo achieves the same accuracy even when the multiple humans performing the motion are as close as 2 feet away from each other, which to the best of our knowledge, no prior RF based motion tracing system has demonstrated.

2 Related Work

Fine-grained motion tracing: Vision based systems such as [51, 4] make use of depth sensors (e.g. Kinect) and infrared cameras (e.g. Wii) to trace the fine-grained motion of a user and enable applications such as gesture recognition, virtual touch screens etc. WiDeo, on the other hand, unlike solutions based on depth imaging or infrared, does not require line of sight to work.

RF based systems like [43] and sensor based systems like [23, 26] perform accurate motion tracing but require instrumentation of users. However, WiDeo achieves accurate fine-grained motion tracing in a device-free manner.

RF based coarse motion tracking and gesture recognition: Recent work such as WiTrack and others [34, 7, 6, 5] has shown the ability to *coarsely track full body motion* (not fine-grained motion of human limbs) using radio waves. Other approaches like [35, 24, 33, 30, 49, 45] track human motion by using ultra-wide band (UWB) signals. All of these approaches are also device-free, but unlike these systems, WiDeo is the first device-free fine-grained motion tracing system that can accurately reconstruct the detailed trajectory of a user's free-form writing or gesturing in the air, where the motion may only span a few tens of centimeters. Such free-form tracing capability is not supported by prior work in RF based gesture recognition or motion tracking. For example, [34] presents a state-of-the-art WiFi based interface, yet it only supports the detection and classification of a predefined set of nine gestures. Moreover, many of these systems [6, 5, 35, 24, 33, 30, 49, 45] require GHz of bandwidth unlike WiDeo which works with regular WiFi bandwidths.

There have been approaches like [48, 50, 27, 36] which use existing WiFi infrastructure, with no hardware modifications to achieve device-free human localization and coarse motion tracking, they use coarse information about the environment in terms of Received Signal Strength Indicators reported by WiFi NICs and require extensive war-driving. In contrast, WiDeo requires minor changes to existing WiFi/LTE APs, re-uses the spectrum allocated for communication by performing fine-grained motion tracing using reflections of communication signals that would have been sent for data communication anyway.

Motion clustering techniques: WiDeo also builds on theoretical work on motion segmentation, clustering and classification [41]. These works are targeted at vision applications that use visible light, and deal with taking a collection of pixels that represent the motion and understanding the underlying motion that occurred. WiDeo on the other hand has to deal with RF signal reflections which pose unique challenges such as multiple reflections, noisier measurements and compact, limited sensors (antenna arrays).

Indoor Localization: A large body of work, ranging from classic RSSI based techniques [15, 9, 47, 37] to recent antenna array based techniques [46, 25, 20] exploit already available WiFi infrastructure to provide indoor localization services for radios. They achieve impressive localization accuracy of a few decimeters. Another line of approaches uses single moving antenna to simulate an antenna array [29]. However WiDeo differs from all of them in two fundamental respects. First, it precisely traces fine-grained motion, rather than a static location. Second, its device-free, the traced object does not need to have any RF transmitters on them.

3 Design

WiDeo's goal is to achieve accurate device-free motion tracing of moving objects. To realize this, WiDeo, like standard ToF camera, incorporates four main components:

Flash: This is the light source used to light up the scene; in WiDeo, this is simply the transmission that the AP in which WiDeo is housed is sending for standard communication. In other words, wireless transmissions used for communicating packets act as the flash for the WiDeo.

Backscatter Sensor: This component looks at the backscatter arising from the environment when the AP's transmission gets reflected and arrives back at the AP. The sensor teases out the individual signals emanating from each reflector in the environment as well as estimates each reflection's intensity, angle of arrival and relative time of arrival. The corresponding component in a standard camera are the image sensors which capture the light (aka the backscatter) from objects in the scene and form a picture of the scene.

Declutterer: The captured backscatter contains a lot of reflections from static objects which act as clutter to the reflections originating from the moving object WiDeo wishes to trace. The declutterer component figures out which of the reflections are from objects WiDeo doesn't care about and eliminates them so that motion tracing can focus only on reflections from moving objects.

Motion Tracer: This component looks at the reflections coming from moving objects over time to predict the actual physical motion that could have produced that sequence of reflections.

We omit the description of the flash component since that is a standard AP transmitter. We describe each of the other three components in detail next. For now assume that the AP's receiver can listen to all the reflections from the environment even though it is transmitting at the same time; we describe how we leverage recent work on full duplex to tackle that challenge in § 3.2.2.

3.1 Backscatter Sensor

The sensor's main challenge is to estimate the parameters of each reflection that makes up the received signal. The reflected signals from these reflectors arrive at the AP with different times of flight (ToF), amplitudes and angles of arrival (AoA), but the receiver only obtains the sum of the signals. Let's assume L reflectors are present and that each reflector k applies an unique unknown distortion function $f_k(x(t))$ to the transmitted signal $x(t)$. So the overall backscatter signal $y(t)$ that is arriving back at the AP can be simply written as:

$$y(t) = \sum_{k=1}^{L} f_k(x(t)) \tag{1}$$

The backscatter sensor's goal is to estimate these functions f_k and then calculate the ToF, amplitude and AoA of the signals reflected from each of these reflectors. As written, the above equation 1 appears intractable, all we know is the transmitted signal $x(t)$ and the overall received signal $y(t)$. How might we tease out the individual reflections? WiDeo makes two novel observations to solve the above under-constrained problem:

Reflector Sparsity: First, WiDeo posits based on recent empirical evidence [16, 19] that the number of significant reflectors in an indoor environment are limited. While there may be many objects, the ones that actually produce sufficiently strong reflections to be visible in the 40 dB of effective dynamic range, which is typical in WiFi radios, are not so numerous. This phenomenon has been extensively proven in empirical wireless communication studies that study the performance of MIMO which critically depends on the number of independent reflectors in an environment [16, 19]. In WiDeo's case, this means that the number of reflectors that could have contributed significantly to the overall signal is limited.

Narrowband Transformations: The second key observation is that WiDeo uses narrowband communication signals and radios as the flash/light source. By narrowband we mean that the signals generated or received by the WiDeo device (the AP) are filtered to only let the signal within the bandwidth, which conforms to FCC regulation, used for communication to come through. For example, if we are using the WiFi channel of width 40MHz at center frequency 2.42GHz, then a passband filter of width 40MHz centered at 2.42GHz is applied at the transmitter and the receiver. Filtering by a passband filter can be modeled as convolution with a sinc pulse of the same bandwidth in the time domain [1]. So the reflected signal (after including the attenuation and delay) is now convolved with a sinc pulse. So the signal that arrives back from a single reflector is actually given by:

$$f_k(x(t)) = \left(\alpha_k x(t-\tau_k)\right) \otimes \mathrm{sinc}(Bt) \tag{2}$$

where B is the communication bandwidth of the signal, α_k is the complex amplitude and τ_k is the overall delay of the reflection or the Time of Flight (ToF) for the k^{th} reflector, and $\otimes$ represents the convolution operator [39]. Eq. 2 can equivalently be written as:

$$f_k(x(t)) = \left(\alpha_k \mathrm{sinc}(B \times (t-\tau_k))\right) \otimes x(t) \tag{3}$$

If there are L reflectors, then all L reflections will undergo different attenuations and ToFs, add up over the air and then get convolved with a sinc pulse. Therefore the overall signal is given by:

$$y(t) = \left(\textstyle\sum_{k=1}^{L} \alpha_k \mathrm{sinc}(B \times (t-\tau_k))\right) \otimes x(t). \tag{4}$$

The sensors now first calculate the overall transformation $h(t)$ that has happened to the transmitted signal $x(t)$, i.e. $y(t) = h(t) \otimes x(t)$ where $h(t)$ is essentially the sum of the transformations applied by all the reflectors. This is classic channel estimation that's used in standard communications (after all every receiver estimates the channel that has transformed the transmitted signal to be able to decode). We refer the reader to the following literature [8] for a review of the different techniques that can be used.

However, WiDeo's problem is quite harder than standard channel estimation which only cares about the overall transformation. Although, WiDeo knows the overall channel $h(t)$, it needs to figure out the amplitudes and time shifts of the sinc pulses that are summed up to produce the overall channel $h(t)$. The equation that WiDeo has to solve is therefore given by:

$$h(t) = \textstyle\sum_{k=1}^{L} \alpha_k \mathrm{sinc}(B \times (t-\tau_k)) \tag{5}$$

We can rewrite the equivalent equation in the digital domain (after all WiDeo will be working in the baseband domain after ADC sampling) as:

$$h[n] = \textstyle\sum_{k=1}^{L} \alpha_k \mathrm{sinc}(B \times (nT_s-\tau_k)), \tag{6}$$

where T_s is the sampling time of ADC. WiDeo's goal is to solve the above equation to determine α_k and τ_k for all reflections.

To tackle this, we now exploit the sparsity observation that the number of significant reflectors in an environment is limited to a handful (typically on the order of 10-15). Specifically, we attempt to find the smallest number (less than 20 in our implementation) of scaled and shifted sinc pulses that could have summed up to produce the overall channel response. Mathematically, we are attempting to solve the following problem:

$$\begin{array}{ll} \min & \sum_n (h[n] - \sum_k \alpha_k \mathrm{sinc}(B \times (nT_s - \tau_k)))^2 + \lambda_r |\alpha|_0 \\ \text{s. t.} & \tau_k \geq 0, |\alpha_k| \leq 1, k = 1:L, n = -N:N. \end{array} \tag{7}$$

Note that the above problem is similar to classic problems in compressive sensing [17, 42, 14]. Like in compressive sensing problem, we are trying to find the minimum number of non-zero components (each component corresponds to a reflector) and the corresponding scaling and shifting factors that best explain the observed channel $h[n]$. The sparsity of the number of reflectors is coerced by the term $\lambda_r|\alpha|_0$, where λ_r is a positive regularization term and $|.|_0$ is the number of non-zero terms in the amplitude vector. However there is one major difference, WiDeo's problem is trying to find the best sparsest combination of parameterized *continuous* basis functions (the sinc pulses parameterized by continuous shift factors), whereas classic compressive sensing is finding the sparsest combination of discrete finite sized vectors that produces some overall vector. We omit the mathematical details here for brevity, but refer the reader to a large body of literature on solving these sparse estimation problems [18, 40, 31]. WiDeo's contribution is to show that the backscatter sensing problem can be formulated using sparsity and compressive sensing intuition.

3.1.1 What if the reflectors are closely spaced?

The above description didn't make any mention of how closely spaced the reflectors are. For example, if the two reflectors are a foot apart, their reflections will arrive at the AP within two nanosecond of each other (wireless signals travel a foot per nanosecond and reflection for objects a foot apart takes 2 nanoseconds). But sampling rates of wireless communication radios are at best around 100Msps (Mega samples per second), which means that two samples are spaced 10ns apart. How could then WiDeo estimate the parameters of the two reflectors that are closely spaced to an order of magnitude closer in time than the sampling period? Even if two reflections are closely spaced in time because their reflectors are almost at the same distance from the AP, they are likely to be at different spatial angles (otherwise they would be the same reflector!). So the spatial dimension provides us the ability to separate reflections in space when they are close in time. The heuristic works in the other direction too, if two reflectors are at the same AoA (because they are on the same radial line), they are likely at different delays and can be separated out.

How do we use this insight to separate out reflections? The intuition is that if the WiDeo AP has an antenna array (typical APs have 4 antennas), then the specific AoA of each reflection imposes a constraint on how the phase of that reflection changes across space. Specifically if the antennas are laid out equidistant at distance d in a straight line, the so called uniform linear array (ULA), and if the AoA is θ, then the relative phase between the signal at any two consecutive antennas is given by $(\phi(\theta) = 2\pi d \sin(\theta)c/\lambda)$, where c is the speed of light in air and λ is the wavelength of the RF carrier. Assuming that there are four antennas in the WiDeo's AP we call the following vector $[0, \phi(\theta), 2\phi(\theta), 3\phi(\theta)]$ of phase differences of all the antennas with respect to the first antenna as the relative phase constraint vector.

In general when more than two backscatter signals are present, each of these backscatter signals arrives at all four antennas, but based on the AoA of these signals the relative phase constraint vectors of these signals will be different. WiDeo uses this insight in the following way. In addition to finding the best sparse signals as described by 7, WiDeo imposes an additional constraint that these estimated sparse solutions should strictly follow the phase vector constraint imposed by the ULA structure leading to the following problem for Ψ antennas:

$$\begin{aligned}
\min \quad & \sum_m \sum_n (h_m[n] - \sum_k \alpha_k e^{-i(m-1)\phi(\theta_k)} \mathrm{sinc}(B \times (nT_s - \tau_k)))^2 \\
& + \lambda_r |\alpha|_0 \\
\text{s. t.} \quad & \tau_k \geq 0, |\alpha_k| \leq 1, k = 1:L, n = -N:N, m = 1:\Psi.
\end{aligned}$$

The $e^{-i(m-1)\phi(\theta_k)}$ term in the optimization objective function is encoding the phase constraint that arises from a specific AoA. In essence, while many signals can fit the time domain constraint given by 7, only few of them can satisfy the relative phase constraint vector thereby further limiting our solution space and hence increasing the accuracy of our estimates despite the closeness of these signals in time. The matching relative phase constraint vector of ULA has one-to-one relationship with AoA, thus using this process we can simultaneously estimate the AoA of the backscatter signals in addition to their amplitude and ToF.

To summarize using the above technique, the sensor outputs a set of reflections with their associated three tuple of parameters: amplitude, ToF and AoA. The next step is eliminating the numerous reflections from static objects that act as clutter to motion tracing problem which we describe below.

3.2 Declutterer

Reflectors abound in the environment and their reflections end up cluttering the backscatter, making it hard for WiDeo to focus on the reflections arriving from the moving object that's being traced. Tracing accuracy can be greatly improved if this clutter can be eliminated. There are two kinds of clutter in decreasing order of harmfulness. The first are reflections from objects nearby whose relative strength w.r.t. to the moving object reflection is greater than the dynamic range of the WiDeo receiver. In this case, the reflection from the moving object is completely lost in the quantization noise and motion cannot be traced. The second is clutter whose strength is within the dynamic range relative to the moving object's reflection. Here information is not lost, but it becomes harder for the tracing algorithm to recover the original motion. WiDeo's declutterer handles both kinds of clutter and eliminates them. We start by describing how to handle the second kind of clutter by guessing which reflections are from moving objects, and then describe how to eliminate the rest of the clutter including the nearby reflectors.

3.2.1 Eliminate Reflections of Static Objects

WiDeo uses a heuristic to loosely identify reflections that are likely to have come from moving objects. The basic idea is to look across sequences of backscatter sensor measurements as shown in Fig. 2, and then make an association of which reflections haven't changed in value and which have. The idea is that the reflections that have continuously changed their parameters (their amplitude, AoA and ToF) will include reflections from moving objects. Everything else is classified as static clutter that has to be eliminated.

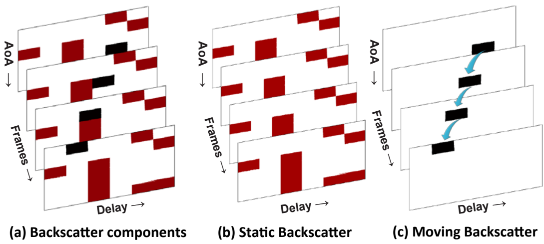

Figure 2: *The figure represents backscatter components obtained from a simulated hand movement in a typical indoor scenario using ray tracing software [2]. The backscatter components collected in each time interval are presented as an image snapshot. The horizontal and vertical axes correspond to ToF and AoA respectively. Each colored pixel corresponds to a backscatter component. Different snapshots stacked one over the other correspond to set of backscatter components obtained in consecutive time intervals. The majority of backscatter components are contributed by static environment, which are shown in the same color to provide contrast with moving backscatter.*

The key question then is to look at snapshots of backscatter over time, associate the backscatter parameters that we believe are coming from the same reflector and then apply the above heuristic . Each snapshot is made up of as many backscatter points as number of reflections, and each point is associated with a three tuple of amplitude, ToF and AoA. WiDeo keeps track of a moving window of such backscatter snapshots (in our current implementation the last 10 snapshots are maintained). The first step is to associate points which are generated from the same reflector between every two successive snapshots, even if the reflector moved between those two snapshots. To do so, we invent a novel point association algorithm across snapshots based on minimizing the amount of change between consecutive snapshots.

Identification of Static Reflections: The algorithm starts by calculating the pairwise distance between every pair of backscatter points in successive snapshots. Distance is defined as the absolute difference in the three parameters (amplitude, ToF and AoA) squared and summed after appropriate normalization. Note that this metric is calculated for all pairs of points, so there would be n^2 distances where n is the number of backscatter points in a snapshot. The goal is to figure out the specific pairings where points in each pair of snapshots are generated by the same backscatter reflector.

Our key insight is that for static objects, the points corresponding to backscatter reflections from that static object in successive snapshots should be at zero distance with respect to each other because by definition they did not move and the associated parameters did not change. Further even for the points that correspond to moving reflectors, given how slow human motion is relative to the length of a backscatter snapshot (a millisecond), the distance between points in successive snapshots that correspond to the moving object is small. So if we can pair the points up such that the overall sum of the distance metrics for these paired points is the minimum among all possible pairings, then very likely we will have associated the right sets of points together.

How do we determine the right point association between successive snapshots? This is a combinatorial assignment problem where we first pass distances between all pairs of points as the input and then pick the set of pairs that minimize the overall sum of distance metric among them. A naive algorithm would be to enumerate all possible assignment of point pairs, which would require evaluation of $n!$ assignments for snapshots with n points. To reduce the complexity, we turn to a classic algorithm in combinatorial optimization known as the Hungarian algorithm [28] which runs in polynomial time. We omit a full description of the algorithm for brevity however, the algorithm is best visualized in terms of a bipartite graph $G = (F1, F2, E)$, where points from the first snapshot are vertices in the set $(F1)$ and points from the second snapshot are in the set $(F2)$ and the edge set (E) consists of all possible edges between vertices in the two sets. The weight of each edge is the distance metric between the backscatter parameters corresponding to the

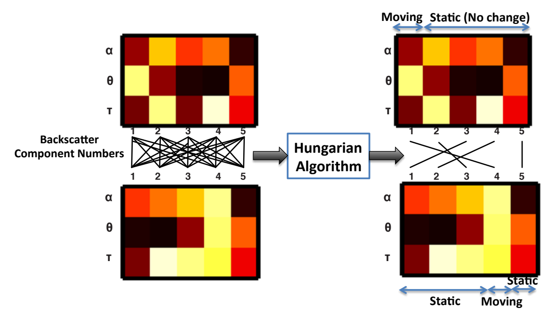

Figure 3: *This figure illustrates application of Hungarian algorithm for a subset of backscatter components obtained in the experiment narrated in Fig. 2. The left side represents the backscatter components in two successive snapshots. The color of each pixel is a representation of the value of* α*(power in dBm),* θ*(in degrees), or* τ*(in ns) according to the appropriate row. We design a distance metric between each component in the first (top) snapshot and each component in second (bottom). The distance thus obtained are represented as edges with appropriate weights (not shown in the figure for clarity). We want to find the matching with minimum weight in the above bipartite graph. Applying Hungarian algorithm results in the least weight matching presented on the right, thus providing a way to associate backscatter components in the two snapshots.*

points the edge connects. The goal of our algorithm is to find a matching with minimum cost, as shown in Fig. 3.

Eliminating Static Clutter: The next step is to eliminate the clutter caused by static backscatter reflectors. This step immediately follows from the above computation; we identify the pairs of points whose distance metric is close to zero, and stays so for at least a fixed number of snapshots (typically around 10 snapshots corresponding to those reflectors being static for 10ms). When we find such points, we declare them to be part of the static clutter. These points are then eliminated from the snapshots and the only points left are those that the algorithm believes to be coming from moving reflectors.

The static clutter elimination step also naturally provides the detection of a new motion that is starting. For example, let's say we start with a completely static environment; in the steady state the declutterer block won't report any parameters because eventually all of the components will be declared static and eliminated as clutter. When a new motion starts and generates new backscatter components, the sensor will report these parameters to the declutterer which will classify them as moving points. Such points are grouped together and passed to the motion tracing block, described in section 3.3, as a new moving object that needs to be traced.

3.2.2 Eliminating Clutter from Nearby Reflectors

In many scenarios, we may have a nearby reflector that is producing strong reflections. If these reflections are stronger than the reflections from the moving object that WiDeo wishes to trace by more than the dynamic range of the radio, all information about the moving object will be lost in the quantization error of the ADC at the receiver. Further remember that WiDeo aims to listen to reflections from the environment while the WiDeo AP is transmitting signals for communication. The transmitted signal also directly leaks through to the receiver and causes interference.

WiDeo's observation is that such clutter is essentially a form of self-interference, and recent work on full duplex radios can be used to eliminate such clutter [12]. Full duplex radios have to solve a similar problem, they have to cancel their own transmitted signal's leakage and reflections that arrive back at the receiver. This self-interference also incorporates reflections from the environment, and recent work has developed sophisticated interference cancellation techniques that can eliminate the self-interference to the noise floor [12]. WiDeo leverages this work. We provide a brief description below, but refer the readers to [10] for a detailed description. WiDeo's contribution is showing how full duplex can be used to build imaging applications rather than the communication applications that full duplex research has focused on.

Conceptually, full duplex radios consist of a programmable canceler component that consists of both analog and digital cancelers. The canceler's main component is a programmable filter which attempts to model the distortions that the transmitted signal goes through before arriving back at the same radio's receiver as self-interference. The canceler takes the transmitted signal as input, passes it through the programmable filter, and then subtracts the filtered signal from the received signal to completely eliminate self-interference.

Note that in traditional full duplex radios, the goal is to completely eliminate the self-interference. WiDeo however is different, in fact some of the self-interference may be coming from moving objects that we do not want to cancel since we want to infer the motion from them. So WiDeo implements a novel modification to traditional full duplex self-interference cancellation. It uses the backscatter sensor measurements to program the filter to only model the *static and strong* reflectors that act as clutter, but intentionally leaves the components that would also have modeled the moving reflectors out. WiDeo figures out which backscatter components correspond to moving reflectors using the static clutter detection algorithm described in the previous section. Thus cancellation is selectively applied only to the static and strong clutter components. Specifically the programmable canceler filter is tuned to implement the following response:

$$h_{c_m} = \sum_{k=1}^{L'} \alpha_k e^{-i(m-1)\phi(\theta_k)} \mathrm{sinc}(B \times (t - \tau_k)) \quad (8)$$

where $\alpha_k, \tau_k, \theta_k$ is the amplitude, ToF and AoA parameters for all L' unwanted reflectors, and h_{c_m} is the response of cancellation filter attached to the mth antenna.

This completes the design of the declutterer component. At this point we have a set of snapshots with points that correspond to moving objects. Further points in successive snapshots are associated with each other if they belong to the same moving reflector. However note that this does not mean we have traced the original moving object itself, all we have isolated is the multiple backscatter reflections from it. The next step is to trace the original object and its motion which produced the snapshots with the moving points.

3.3 Tracing the Actual Motion

Each WiDeo AP sends the isolated backscatter measurements arising from moving objects it computes from the previous step to the central server. Whenever a new motion starts, its quite likely that many of the WiDeo APs will detect the backscatter measurements from this new motion. The server collects backscatter snapshots over a period of 10ms from all participating radios, and assumes that any moving backscatter detected by any of the radios are coming from the same object. The heuristic implicitly makes the assumption that two new and independent human motions won't start within an interval of 10ms. Given the timescales at which human motion happens, 10ms is a negligible amount of time and we believe that such asynchrony is very likely in practice. Note that this does not mean that two independent motions cannot be occurring simultaneously, we are only making the assumption that they don't start within 10ms of each other.

3.3.1 Localizing the origin of the motion

The first step the server implements is actually to localize the origin of the motion that just started. The server has measurements from multiple radios across multiple snapshots, and very likely the new motion will be detected at many of these radios. So, how might we estimate the location of the new motion? The idea is that the measurements collected at the WiDeo APs imposes constraints on where the moving reflector is located. We demonstrate the idea using the AoA measurement. Let's say the locations of the M WiDeo APs involved in motion tracing are given by $(x_i, y_i), i = 1, \ldots, M$. Similar to many other state-of-the-art localization systems [46, 38] using WiFi, the locations of the APs (or the anchors) are assumed to be known in advance. Let the AoA measurements of the reflector at the APs be denoted by $\theta_i, i = 1, \ldots, M$ and the current estimate of the object's location is (x, y). So the most likely location of the object is one that minimizes the following metric:

$$\begin{array}{ll} \min & \sum_{i=1}^{M} (\bar{\theta}_i + b_{\theta_i} - \theta_i)^2 \\ \text{s. t.} & \bar{\theta}_i = \text{AoAULA}((x, y), (x_i, y_i)) \end{array} \tag{9}$$

The above equation is stating the fact that the predicted angle of arrival at each of the WiDeo APs given the estimated location of the reflector and the location of the APs must closely match the actual AoA measured by each of the APs. The function $\text{AoAULA}((x, y), (x_i, y_i))$ computes the AoA seen by the tracing radio located at (x_i, y_i) from a reflector located at (x, y). However there is a new factor b_{θ_i} that represents the bias to model multipath reflections. This is because the moving backscatter not only corresponds to the direct backscatter from the object but also the backscatter from the reflections of the backscatter. For example, if a backscatter reflection from a moving object is further reflected by a wall before arriving at the AP, the ToF parameter will have a constant bias that reflects the extra time it takes to traverse the extra distance corresponding to going to the wall and reflecting off it. Similar bias exists for both the amplitude and AoA measurements. Further the bias values are unknown and hence are a variable in the optimization. The value of (x, y) that minimizes the above metric is likely the best estimate of the location of the reflector.

We can also use other parameters like ToF and power to estimate the location of the target. In our actual implementation, we solve a more sophisticated optimization problem than the simple optimization problem in 9. Specifically, WiDeo uses AoA, ToF, and backscatter signal strength measurements over multiple frames for the particular backscatter, say J frames, and declares the origin of the motion as the location that minimizes the following objective function described by Eq. 10

$$\sum_{j=1}^{J} \sum_{i=1}^{M} [(\bar{\alpha}_i - \alpha_{ij})^2 + (\bar{\tau}_i + b_{\tau_i} - \tau_{ij})^2 + (\bar{\theta}_i + b_{\theta_i} - \theta_{ij})^2], \tag{10}$$

where α_{ij}, τ_{ij}, and θ_{ij} are the power, ToF, and AoA respectively of the backscatter observed by the i^{th} AP in the j^{th} frame and the variables $\bar{\alpha}_i$, $\bar{\tau}_i$, and $\bar{\theta}_i$ are the values of respective backscatter parameters that would have been observed at the APs if the object was actually located at that particular location. We follow a simple path loss model [21] to describe the relation between the location of the object and the backscatter signal strength $\bar{\alpha}_i$. The variables b_{τ_i} and b_{θ_i} represent the bias in ToF and AoA respectively due to reflections of the backscatter from the object. This problem of minimizing Eq. 10 is non-convex, therefore we apply a widely used heuristic known as sequential convex optimization to solve it [13].

We note that Eq. 9 as such is an ill-posed problem without a unique solution because each AP introduces its own bias terms for backscatter parameters. However, in Eq. 10, by collecting measurements over enough number of backscatter frames, the number of measurements become greater than the number of variables and the optimization problem becomes well-posed. Further the parameters of simple path loss model used to model $\bar{\alpha}_i$ are also estimated as part of the minimizing Eq. 10 and need not be known ahead of time.

3.3.2 Tracing Motion

Once the newly detected moving object is localized, the next step is to trace the object's motion as it moves and produces new measurements via our backscatter sensor. Remember that the new measurements are naturally associated with the measurements from the previous snapshots via the declutterer described in § 3.2. So the algorithm has already clustered backscatter measurements coming from the same moving reflector together, and we can operate the motion tracing algorithm on each cluster of measurements separately. Hence we describe the tracing algorithm as if there is a single motion occurring and a single set of backscatter measurements being produced from it across successive snapshots.

Our approach to this problem is to build a dynamical model about the motion that is occurring and progressively refine its parameters. There are several parameters to the motion model: current position of the object, velocity, direction of motion, acceleration, and bias in each backscatter parameter due to the indirect reflections. Both the bias and initial position variables are initialized using the output of the localization algorithm in the previous step. Velocity, acceleration and direction of motion are initially set to zero and then updated over time as new measurements come along. Note that we also allow the bias parameter to change over time, after all as the object moves, the bias for each parameter changes.

The key insight is as follows: at every point in the traced motion, given the estimate of the motion model at that instant, WiDeo can predict what the backscatter sensor measurements for that moving reflector should be (given the estimate for the locations of the reflector and the WiDeo AP and the biases in the parameters we can calculate the expected amplitude, ToF and AoA of the reflections). WiDeo also of course has access to the actual backscatter sensor measurements at that instant for the same moving reflector, so we can calculate the error between the predicted and the actual backscatter measurement. The goal of the motion tracing component is to *minimize the sum of these backscatter prediction errors over the entire motion trajectory in a sequential fashion*. The algorithm proceeds in three steps at each time instant:

Model based prediction: In this step, WiDeo calculates the new position of the reflector given the previous position and motion model parameters namely, velocity and acceleration. It then uses this extrapolated position along with the estimates of the bias for the backscatter parameters to calculate what the new values of the amplitude, ToF and AoA of the reflection should be.

Backscatter prediction error computation: Compute the difference between the above predicted and measured backscatter parameters.

Model update from error: Update motion model parameters such that the overall backscatter prediction error is minimized across the entire trajectory. The update step uses a classic technique in dynamic estimation: the Kalman filter [44]. Kalman filter theory shows that assuming the measurement noise and motion modeling error is Gaussian, the update is dependent on two factors. First factor is the size of the prediction error itself, i.e. if the error is large then a larger update to the model is required and vice-versa. The second factor is a gain term that modulates this error term. The gain factor is chosen such that the accumulated error between all the observations of the measured backscatter parameters so far and the best prediction that the motion model can make is minimized. In essence the gain signifies the effect of accumulated errors in the motion model, for example if the measurements are noisy the gain should be chosen small to account for the unreliable nature of the measurement and vice-versa. We omit the proof and refer the readers to [44] for a more detailed mathematical treatment of the Kalman filter and how to compute the gain factor given the motion model and history of backscatter measurements and prediction errors.

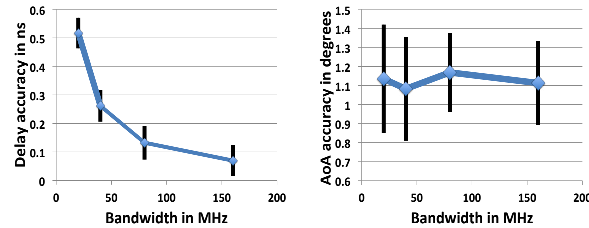

Figure 4: *Accuracy of WiDeo's algorithms in estimating the delays and AoAs of backscatter. WiDeo achieves an accuracy of 300ps and* 1.2 *degrees (with error bar representing standard deviation) at 40MHz bandwidth used in WiFi signals.*

The convergence of the motion tracer takes a few snapshots, after this point it constantly updates its motion model parameters. Reconstructing the motion is now akin to starting with the initial point and performing a directional piece-wise integration using the speed and direction of motion parameter at each time step. An instance of the above algorithm is executed for each detected motion.

4 Evaluation

We implement a prototype of WiDeo using the WARP software radios using WiFi compatible OFDM PHY with a bandwidth of 20MHz at 2.4GHz. The radio is set up to use 4 antennas and all RX chains are phase synchronized like in a MIMO radio. The spacing of the antennas is $\lambda/2$ and the overall width of ULA is 18cm. The declutterer is designed using analog cancellation circuit boards based on the design described in [11]. From the time it receives information about the clutters to be canceled, the declutter takes few microseconds to remove their effect and improve the dynamic range. The optimization algorithms that measure the backscatter parameters and the rest of the tracing algorithms are implemented in a host PC in C using the cvxgen toolbox [32] and Matlab. Although the current implementation of WiDeo is not realtime we believe it is possible with a few architectural changes and speed optimizations in the future.

4.1 Back-scatter sensor benchmarks

We start with micro-benchmarks of the backscatter sensor that underpins motion tracing. The goal here is to demonstrate that WiDeo's backscatter measurement algorithms provide high accuracy and fine resolution.

Accuracy: We first measure WiDeo's accuracy in measuring backscatter parameters. Given the complex geometry of indoor environments, a natural question is how do we know ground truth for all the multipaths to evaluate the accuracy of WiDeo? We perform controlled experiments by connecting the RX chains with wires from the transmitted chain. The lengths of the wires are varied to provide different delays, attenuators on each wire provide tunable amplitude, and phase shifters are introduced to simulate AoA. This wired experiment can create 10 different backscatter components. To mimic realistic indoor reflections, we vary the lengths and attenuations by sampling it from an indoor power delay profile [19], and AoA is picked uniformly at random. We vary the receiver bandwidth from 20MHz to 160MHz.

Since WARP radios can only support up to 40MHz bandwidth, we use signal analyzers for the higher bandwidth experiments. Higher receiver bandwidth is expected to help improve accuracy because we are getting finer-grained observations in time due to higher sampling rates. However, the default configuration for WiDeo unless stated otherwise is WARP radios with 20MHz bandwidth.

Fig. 4 plots the overall estimation accuracy for delay and AoA of the backscatter components as a function of bandwidth, we omit amplitude results for brevity (their accuracy was within 1dB). As we can see WiDeo provides extremely high accuracy, measuring delay to within 0.3ns accuracy for a bandwidth of 40MHz, the most commonly used WiFi bandwidth. Further AoA accuracy is 1.2 degrees at 40MHz bandwidth. Accuracy improves slightly for delay estimation with bandwidth, which is expected since we now get more closely spaced samples that helps discern delay better. AoA accuracy is not affected much by bandwidth since that is primarily determined by the number of antennas.

Resolution: Next we conduct an experiment to measure WiDeo's resolution, i.e. how close two backscatter reflectors can be before WiDeo's algorithms fail to disambiguate their respective parameters? First, we create two backscatter components whose delays are far from each other by using wires of different lengths. We then slowly decrease the relative delay and measure at what relative delay the accuracy is a factor of two worse than in Fig. 4. Next we repeat the same experiment, but instead of delay, we make the AoA of two backscatter components very close to each other and check at what relative AoA the accuracy is a factor of two worse than in Fig. 4. The results are presented in Fig. 5.

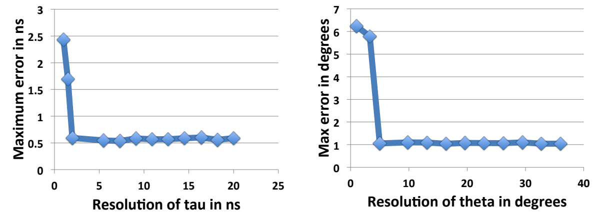

Figure 5: *WiDeo can accurately measure parameters even when the backscatter are spaced only 2ns apart in time or 5 degrees apart in spatial orientation.*

WiDeo can resolve delay to 2ns, distinguishing two gesturing humans separated by only one foot. WiDeo can resolve angle to 5 degrees, distinguishing humans 1 foot apart at 12 feet away.

Range and Dynamic Range: A third benchmark is how weak a backscatter signal can be before it cannot be estimated by WiDeo. Clearly if a backscatter is weaker than the noise floor of the receiver radio (-90dBm), then WiDeo cannot detect it. But how much above the noise floor does the backscatter have to be for accurate measurement? We repeat the accuracy experiment shown in Fig. 4 by picking the parameters for 9 components from the power delay distribution while progressively decreasing the strength of the 10th backscatter component.

Fig. 6 (on left) plots estimation accuracy of different backscatter parameters as a function of the received strength at the radio. When the backscatter component is weaker than -70dBm (i.e less than 20dB above the noise floor of the receiver), WiDeo's accuracy degrades to around 6ns for the delay. In practice this means that the motion that is being traced needs to happen within 16 feet radius of the radios for high backscatter sensor accuracy. Note that the range of motion tracing can be more than 16 feet as motion tracing may not need parameters to be highly accurate.

Another related benchmark is the resilience of WiDeo in scenarios where there is backscatter from a nearby reflector and the motion we actually want to trace is farther away and producing weak backscatter. To test this we conduct a controlled experiment where there are two backscatter reflectors, one nearby whose strength is kept constant at 10dBm while the other one is made weaker and weaker. We plot the accuracy of backscatter measurement for the weaker component as a function of the difference in strength w.r.t. the strong backscatter component in Fig 6(on right). WiDeo accurately measures components as weak as 80dB below the strong reflector, well beyond the radio's 40dB dynamic range. This works because the declutterer estimates the strong component, then cancels it completely all the way down to the noise floor.

Note that both the maximum range and the dynamic range of WiDeo is limited by the noise floor of the radio being used and the transmitted power by the sensor, and not due to the limitations of WiDeo's algorithm. This is because WiDeo's cancellation can cancel specified reflections all the way to the noise floor. If the cancellation were imperfect and doesn't reach the noise floor; for example, a 20dB residue will limit WiDeo to sensing signals above -50dBm rather than the -70dBm shown in Fig 6,

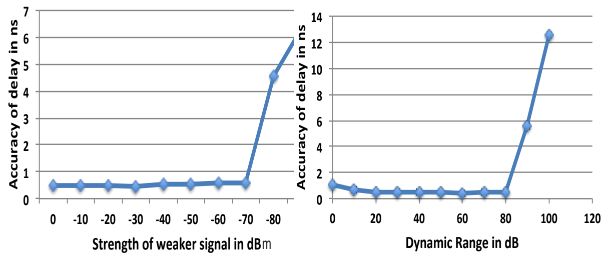

Figure 6: *(on left)WiDeo can accurately estimate backscatter parameters for reflections that are as weak as* -70*dBm. (on right) It can also accurately estimate parameters for very weak backscatter components even when there is a strong backscatter component present which is 80dB stronger.*

which would reduce the range as well to 2 feet.

4.2 Motion tracing benchmarks

We now evaluate WiDeo's ability to accurately trace motion in indoor environments. We calculate two metrics
Location accuracy: This is the accuracy of the localization of a motion that is detected by WiDeo. We use Euclidean distance between the centroid of the ground truth motion and the estimated motion as the metric.
Motion tracing accuracy: This is the accuracy of the traced motion. The metric we used is the root mean square error of the traced motion which we calculate by computing the distance of each point in the traced motion with the ground truth motion trace at that point. The distances are squared and added up and normalized by the number of points before taking the square root. Hence, the metric represents the motion tracing error in meters. Similar to [43], we remove any offset between the ground truth motion and traced motion.

The locations tested for motion tracing accuracy spans all scenarios: non line-of-sight (NLOS) to any of the tracing radios, LOS to a subset of the tracing radios and through walls in an indoor environment spanning 600sq.ft. By default, unless stated otherwise, the number of tracing radios is fixed to three and they are deployed at three fixed but arbitrarily picked locations in the testbed. The motions we trace are actually humans sketching various shapes with their hands. By default, unless stated otherwise, we have two humans performing motion concurrently in our experiments.

We could not find any recent system that implements fine-grained motion tracing within the design requirements of WiDeo: namely being device-free, compact and one that uses existing communication signals and spectrum. RF-IDraw [43] as discussed before is not device-free, nor compact. Other recent work such as WiTrack [6] is device free but implements coarse tracking of the entire human body moving, but cannot track fine-grained motion of human limbs. Hence we refer the reader to § 1,§ 2 for a qualitative comparison to these related systems.

Our experimental results show the following

- WiDeo accurately traces motion, it achieves a median localization accuracy of 0.8m and motion tracing accuracy of 7cm.
- WiDeo can accurately trace multiple independent motions, tracing as many as five independent and concurrent motions with an error less than 12cm.
- WiDeo's resolution is 0.5m, i.e. if the two independent motions are occurring within half a meter or higher of each other, WiDeo can trace them accurately.
- Accuracy improves modestly with the number of radios involved in the tracing. When we increase the number of radios to five, localization accuracy improves to 0.7m, whereas motion tracing accuracy improves to 6cm.

4.2.1 Motion tracing experiments

We use a SPEAG hand [3] to perform motion tracing experiments. This model hand is designed to have same dimensions and absorption/reflection characteristics as that of a typical human hand in 2.4GHz frequency range.

This hand is placed over a chart with figures of different shapes like the one shown in Fig. 7. Several markers are drawn on the shape and the backscatter is captured by WiDeo's APs when the SPEAG hand is placed on each of these markers. The markers are spaced apart by approximately 5cm so as to emulate a scenario where WiDeo collects measurements every 10ms when a human hand is moving at a speed of 5m/s [22]. The ground truth location for each marker on the chart is obtained by using laser range measurements and architectural drawings. In Fig. 7, the shape in the blue shown in the right is found using such laser measurements. By placing the model hand in all the locations of the chart sequentially, we emulate the scenario where a human hand traces the particular trajectory whose ground truth is accurately determined.

We conducted experiments in scenarios with one, two, and all three APs in LOS. WiDeo's accuracy is tabulated in Fig. 8. WiDeo achieves an accuracy of 5.1cm when the APs are in LOS, and is still quite accurate at 12.8cm when two of the APs are in NLOS.

4.2.2 Understanding WiDeo's motion tracing

Because of the time consuming nature of the data collection procedure for the above testbed experiments, we

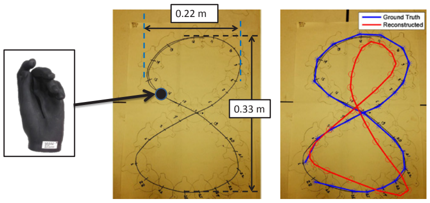

Figure 7: *(Left) A chart with figure of 8 with multiple markings where SPEAG hand (in inset) was placed and the data was captured by WiDeo's AP. (Right) Ground truth data obtained using laser range finder (in blue) along with the motion trace reconstructed by WiDeo (in red) using 3 APs.*

Motion	Localization Accuracy (m)		Motion Tracing Accuracy (cm)	
	Testbed	Wireless InSite	Testbed	Wireless InSite
All AP LOS	.54	.54	5.1	5.3
1AP NLOS	1.1	1.1	8.5	8.4
2AP NLOS	1.61	1.63	12.8	12.5

Figure 8: Median accuracy for different motion shapes obtained using SPEAG hand and Wireless InSite tool.

can only perform a limited number of experiments using it. To extensively test the motion tracing accuracy of WiDeo under more diverse conditions, we simulated the entire system in an electromagnetic emulation environment called Wireless InSite [2]. Wireless InSite is a ray tracing based tool to accurately model RF propagation in any indoor environment with walls and other objects. This tool enables us to emulate complex indoor environments in which WiDeo will be used, as well as know the ground truth for every experiments. To demonstrate Wireless InSite produces similar results, we modeled the testbed described above and then collected data for the same scenarios in Wireless InSite. We emulated the dynamic range and progressive interference cancellation on the data obtained from Wireless InSite simulation. Fig. 8 compares the accuracy of motion tracing achieved with Wireless InSite data with that obtained with the physical experiments. We see that the two results match very closely which is due to Wireless InSite's ability to accurately model indoor RF environments. Hence, in the rest of the sections, we use Wireless InSite to analyze performance of WiDeo in more detail.

4.2.3 WiDeo's motion tracing performance

We now evaluate the WiDeo's motion tracing accuracy by conducting extensive experiments using Wireless InSite. Specifically, we vary the placement of the two moving humans arbitrarily in the testbed across 100 different locations. We calculate the median localization error and the root mean square error of the traced motion. We plot the CDFs in Fig. 9.

WiDeo achieves a median localization error of 0.8m and a median tracing error of 7cm. The tail errors are often cases where the human motion is happening in a dead zone where the backscatter to any of the tracing radios is weaker than -80dBm. In these cases the backscatter measurement itself has worse accuracy which translates to poor accuracy for motion tracing. However WiDeo still achieves a motion tracing accuracy better than 15cm in 90% of the scenarios.

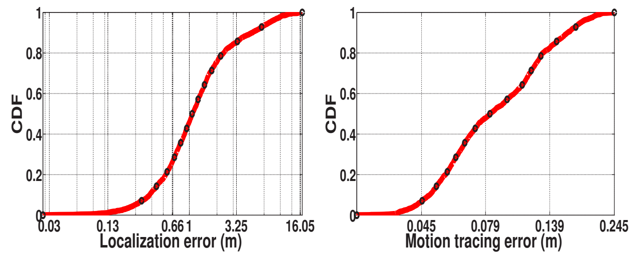

Figure 9: *WiDeo's motion tracing is extremely accurate; it traces fine-grained motion with a median localization error of 0.8m and motion tracing error of 7cm.*

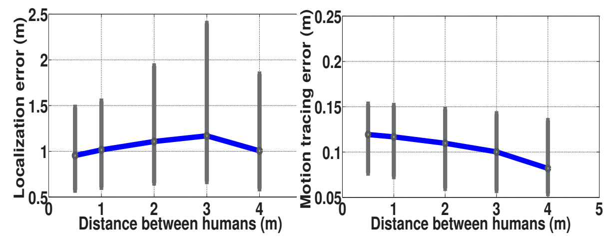

Figure 10: *WiDeo provides high resolution motion tracing, it can accurately trace two independent motions occurring even if they are only spaced 0.5m apart (with error bar representing standard deviation).*

4.2.4 Resolution

Many applications that might build upon WiDeo's motion tracing capability care about resolution, i.e. how close can two independent human motions be occurring and WiDeo can still trace them accurately (e.g multi-player video games). To conduct this experiment we progressively move the two moving humans closer to each other and plot the worse of the two motion tracing accuracies as reported by WiDeo in Fig. 10.

WiDeo achieves a motion tracing resolution of 0.5 meters while still achieving an extremely good tracing accuracy of 12cm. So two humans could be standing a little bit more than a foot away from each other (e.g. in a video game), moving their hands closest to each other simultaneously, and still be able to accurately trace their motion.

We also observed that localization error is unaffected; the error is the same as in Fig. 9 . This is expected since the localization technique works by combining measurements from multiple tracing radios when a new moving backscatter component is detected. Since we assume that two human motions do not start exactly at the same time and are usually spaced at least 10ms apart, WiDeo's localization algorithms have a sufficiently long window of time (10ms) in which they can perform localization on a single new object without the presence of a nearby moving object. The same argument applies when the second new motion is detected, by then the first motion is localized and can be accounted for and localization can focus only on the new backscatter components that arise from new moving object.

4.2.5 Impact of number of tracing radios

In this experiment, we see impact on accuracy as we vary the number of radios performing tracing in WiDeo. We conduct the same experiment as in § 4.2.3, but vary the number of tracing radios from one to five. We plot five different CDFs of localization and motion tracing error in Fig. 11 .

As we can see, WiDeo's localization error is poor (4m) with a single tracing radio. This is expected, since WiDeo relies on triangulation to localize well. However motion tracing error is less affected, WiDeo still traces with less than 12cm error. Consequently while we cannot localize with a single radio, we can still trace. The reason is that with a single tracing radio, we cannot get an accurate estimate of the depth (location), but the relative motion from that initial location can still be accurately traced since it only depends on relative shifts in backscatter measurements, which are quite accurate.

Increasing the number of tracing radios helps with localization error, it goes down to 0.7m with five tracing radios. Motion tracing, which is already quite accurate even with a single radio, improves slightly to 7cm. This is expected since triangulation improves with more radios and hence localization improves. However backscatter measurement doesn't depend on having multiple observations, it's done independently by each radio. Hence tracing accuracy only improves by a small amount.

4.2.6 Distinct motions that WiDeo can trace

In this experiment, we check how many independent concurrent human motions can WiDeo trace. We vary the number of human motions occurring concurrently from one to six and plot the median tracing accuracy in Fig. 12.

WiDeo can trace up to five concurrent motions with an accuracy of 12cm. To the best of our knowledge, no prior WiFi based system has demonstrated being able to trace five moving humans concurrently. Beyond that accuracy worsens. The reason is that there aren't enough radios to

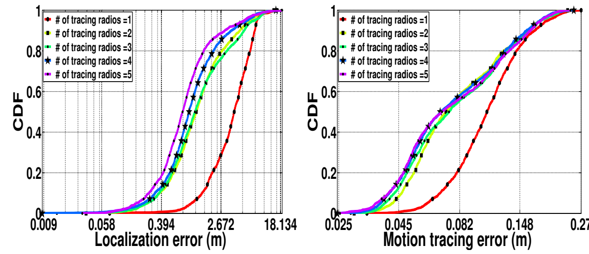

Figure 11: *WiDeo's localization accuracy improves with the number of tracing radios to 0.7m because of better triangulation. However tracing accuracy is unaffected because WiDeo's algorithm's can trace accurately even with information from a single tracing radio.*

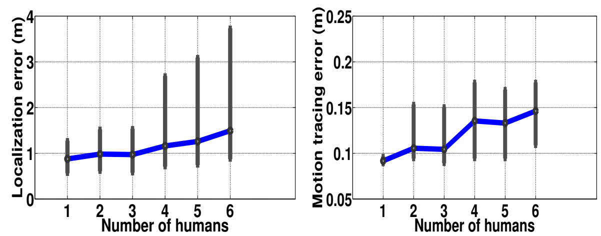

Figure 12: *WiDeo can accurately trace as many as five independent motions that are occurring simultaneously (with error bar representing standard deviation).*

provide sufficient number of backscatter measurements to disentangle these motions. Being able to trace five concurrent motions is sufficient for a home environment, but not for work environments where a far greater amount of motion is expected.

5 Conclusion

This paper demonstrated the surprising capability to build motion tracing camera using WiFi signals as the light source. The fundamental contributions are algorithms that can measure WiFi backscatter and mine them to trace motion. We plan to prototype many interesting applications that builds on top of WiDeo, including gesture recognition, indoor navigation, elderly care and security applications.

References

[1] *David Tse , Fundamentals Wireless Communications.* `http://www.eecs.berkeley.edu/~dtse/Chapters_PDF/Fundamentals_Wireless_Communication_chapter2.pdf`.

[2] *Modeling Indoor Propagation.* `http://www.remcom.com/examples/modeling-indoor-propagation.html`.

[3] *SPEAG Hand*. http://www.speag.com/products/em-phantom/hand/sho-v2-3rb-lb/.

[4] *Wii*. http://en.wikipedia.org/wiki/Wii.

[5] ADIB, F., KABELAC, Z., AND KATABI, D. Multi-Person Motion Tracking via RF Body Reflections.

[6] ADIB, F., KABELAC, Z., KATABI, D., AND MILLER, R. C. 3D Tracking via Body Radio Reflections. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)* (Seattle, WA, Apr. 2014), USENIX Association, pp. 317–329.

[7] ADIB, F., AND KATABI, D. See Through Walls with WiFi! In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (New York, NY, USA, 2013), SIGCOMM '13, ACM, pp. 75–86.

[8] ARSLAN, H., ET AL. Channel Estimation for Wireless OFDM Systems. *IEEE Surveys and Tutorials 9*, 2 (2007), 18–48.

[9] BAHL, P., AND PADMANABHAN, V. RADAR: an in-building RF-based user location and tracking system. *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064) 2* (2000), 775–784.

[10] BHARADIA, D., JOSHI, K. R., AND KATTI, S. Full Duplex Backscatter. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks* (2013), ACM, p. 4.

[11] BHARADIA, D., AND KATTI, S. Full Duplex MIMO Radios. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)* (Seattle, WA, Apr. 2014), USENIX Association, pp. 359–372.

[12] BHARADIA, D., MCMILIN, E., AND KATTI, S. Full Duplex Radios. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (New York, NY, USA, 2013), SIGCOMM '13, ACM, pp. 375–386.

[13] BOYD, S., AND VANDENBERGHE, L. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[14] CANDES, E., AND ROMBERG, J. Sparsity and Incoherence in Compressive Sampling, 2006.

[15] CHINTALAPUDI, K., PADMANABHA IYER, A., AND PADMANABHAN, V. N. Indoor Localization Without the Pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (2010), ACM, pp. 173–184.

[16] CZINK, N., HERDIN, M., AND ZCELIK ERNST BONEK. Number of Multipath Clusters in Indoor MIMO Propagation Environments.

[17] DONOHO, D. L. Compressed sensing. *IEEE Trans. Inform. Theory 52* (2006), 1289–1306.

[18] EKANADHAM, C., TRANCHINA, D., AND SIMONCELLI, E. P. Recovery of sparse translation-invariant signals with continuous basis pursuit. *Signal Processing, IEEE Transactions on 59*, 10 (2011), 4735–4744.

[19] ERCEG, V., SCHUMACHER, L., KYRITSI, P., AND ET AL. TGn channel models. *Tech. Rep. IEEE P802.11, Wireless LANs, Garden Grove, Calif, USA* (2004).

[20] GJENGSET, J., XIONG, J., MCPHILLIPS, G., AND JAMIESON, K. Phaser: Enabling Phased Array Signal Processing on Commodity WiFi Access Points. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2014), MobiCom '14, ACM, pp. 153–164.

[21] GOLDSMITH, A. *Wireless communications*. Cambridge university press, 2005.

[22] GUPTA, S., MORRIS, D., PATEL, S., AND TAN, D. Soundwave: using the Doppler Effect to Sense Gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 1911–1914.

[23] HARRISON, C., TAN, D., AND MORRIS, D. Skinput: Appropriating the Body as an Input Surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), ACM, pp. 453–462.

[24] JIA, Y., KONG, L., YANG, X., AND WANG, K. Through-wall-radar localization for stationary human based on life-sign detection. *2013 IEEE Radar Conference (RadarCon13)*, 3 (Apr. 2013), 1–4.

[25] JOSHI, K., HONG, S., AND KATTI, S. PinPoint: Localizing Interfering Radios. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation* (Berkeley, CA, USA, 2013), nsdi'13, USENIX Association, pp. 241–254.

[26] KIM, D., HILLIGES, O., IZADI, S., BUTLER, A. D., CHEN, J., OIKONOMIDIS, I., AND OLIVIER, P. Digits: Freehand 3D Interactions Anywhere Using a Wrist-Worn Gloveless Sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (2012), ACM, pp. 167–176.

[27] KOSBA, A., SAEED, A., AND YOUSSEF, M. RASID: A robust WLAN device-free passive motion detection system. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on* (March 2012), pp. 180–189.

[28] KUHN, H. W. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly 2* (1955), 83–97.

[29] KUMAR, S., HAMED, E., KATABI, D., AND ERRAN LI, L. LTE radio analytics made easy and accessible. *Proceedings of the 2014 ACM conference on SIGCOMM - SIGCOMM '14* (2014), 211–222.

[30] MAAREF, N., MILLOT, P., PICHOT, C., AND PICON, O. A Study of UWB FM-CW Radar for the Detection of Human Beings in Motion Inside a Building. *Geoscience and Remote Sensing, IEEE Transactions on 47*, 5 (May 2009), 1297–1300.

[31] MALLAT, S., AND ZHANG, Z. Matching Pursuit With Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing 41* (1993), 3397–3415.

[32] MATTINGLEY, J., AND BOYD, S. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering 13*, 1 (2012), 1–27.

[33] NARAYANAN, R. M. Through-wall radar imaging using UWB noise waveforms. *Journal of the Franklin Institute 345*, 6 (Sept. 2008), 659–678.

[34] PU, Q., GUPTA, S., GOLLAKOTA, S., AND PATEL, S. Whole-Home Gesture Recognition using Wireless Signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking* (2013), ACM, pp. 27–38.

[35] RALSTON, T., CHARVAT, G., AND PEABODY, J. Real-time through-wall imaging using an ultrawideband multiple-input multiple-output (MIMO) phased array radar system. In *Phased Array Systems and Technology (ARRAY), 2010 IEEE International Symposium on* (Oct 2010), pp. 551–558.

[36] SEIFELDIN, M., SAEED, A., KOSBA, A. E., ELKEYI, A., AND YOUSSEF, M. Nuzzer: A Large-Scale Device-Free Passive Localization System for Wireless Environments. *IEEE Transactions on Mobile Computing 12*, 7 (July 2013), 1321–1334.

[37] SEN, S., CHOUDHURY, R. R., AND NELAKUDITI, S. SpinLoc: Spin Once to Know Your Location. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications* (New York, NY, USA, 2012), HotMobile '12, ACM, pp. 12:1–12:6.

[38] SEN, S., LEE, J., KIM, K.-H., AND CONGDON, P. Avoiding Multipath to Revive Inbuilding WiFi Localization. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2013), MobiSys '13, ACM, pp. 249–262.

[39] SHEN, Y., AND MARTINEZ, E. Channel Estimation in OFDM Systems. *Application Note, Freescale Semiconductor* (2006).

[40] TIBSHIRANI, R. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B 58* (1994), 267–288.

[41] TIPALDI, G. D., AND RAMOS, F. Motion clustering and estimation with conditional random fields. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on* (2009), IEEE, pp. 872–877.

[42] TROPP, J. A., AND GILBERT, A. C. Signal recovery from partial information via Orthogonal Matching Pursuit. *IEEE TRANS. INFORM. THEORY* (2005).

[43] WANG, J., VASISHT, D., AND KATABI, D. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (New York, NY, USA, 2014), SIGCOMM '14, ACM, pp. 235–246.

[44] WELCH, G., AND BISHOP, G. An Introduction to the Kalman Filter. Tech. rep., Chapel Hill, NC, USA, 1995.

[45] WILSON, J., AND PATWARI, N. See-Through Walls: Motion Tracking Using Variance-Based Radio Tomography Networks. *IEEE Transactions on Mobile Computing 10*, 5 (May 2011), 612–621.

[46] XIONG, J., AND JAMIESON, K. ArrayTrack: A Fine-Grained Indoor Location System. In *NSDI* (2013), pp. 71–84.

[47] YOUSSEF, M., AND AGRAWALA, A. The Horus WLAN Location Determination System. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2005), MobiSys '05, ACM, pp. 205–218.

[48] YOUSSEF, M., MAH, M., AND AGRAWALA, A. Challenges: Device-free Passive Localization for Wireless Environments. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking* (New York, NY, USA, 2007), MobiCom '07, ACM, pp. 222–229.

[49] ZETIK, R., CRABBE, S., KRAJNAK, J., PEYERL, P., SACHS, J., AND THOMA, R. Detection and localization of persons behind obstacles using M-sequence through-the-wall radar, 2006.

[50] ZHANG, D., MA, J., CHEN, Q., AND NI, L. M. An RF-Based System for Tracking Transceiver-Free Objects. In *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications* (Washington, DC, USA, 2007), PERCOM '07, IEEE Computer Society, pp. 135–144.

[51] ZHANG, Z. Microsoft Kinect Sensor and Its Effect. *MultiMedia, IEEE 19*, 2 (Feb 2012), 4–10.