Bringing Gesture Recognition To All Devices

Bryce Kellogg[†], Vamsi Talla[†], and Shyamnath Gollakota
University of Washington
[†]Co-primary Student Authors

Abstract

Existing gesture-recognition systems consume significant power and computational resources that limit how they may be used in low-end devices. We introduce AllSee, the first gesture-recognition system that can operate on a range of computing devices including those with no batteries. AllSee consumes three to four orders of magnitude lower power than state-of-the-art systems and can enable always-on gesture recognition for smartphones and tablets. It extracts gesture information from existing wireless signals (e.g., TV transmissions), but does not incur the power and computational overheads of prior wireless approaches. We build AllSee prototypes that can recognize gestures on RFID tags and power-harvesting sensors. We also integrate our hardware with an off-the-shelf Nexus S phone and demonstrate gesture recognition in through-the-pocket scenarios. Our results show that AllSee achieves classification accuracies as high as 97% over a set of eight gestures.

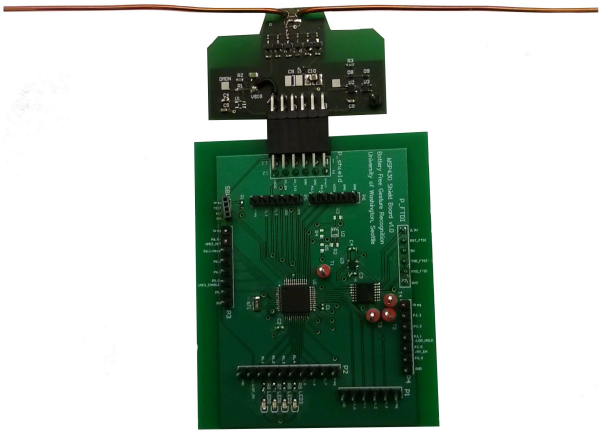

Figure 1: **AllSee Prototype.** It has two main components: a pluggable receiver that extracts the amplitude of wireless signals (e.g., TV and RFID transmissions), and our classification logic implemented on a microcontroller. It also comes with LEDs and a UART interface.

1 Introduction

There is growing interest in using human gestures as a means of interaction with computing devices. The success of Xbox Kinect has led to the development of novel gesture-recognition approaches including those based on infrared [4], electric-field sensing [6], and more recently, wireless signals (e.g., Wi-Fi) [32].

Existing approaches, however, are limited in how they may be used for low-end devices. For example, the "air gesture" system on Samsung Galaxy S4 drains the battery when run continuously [2]. Similarly, wireless approaches [18, 32] consume significant power and computational resources that limit their applicability to plugged-in devices such as Wi-Fi routers.

This is due to two main reasons: First, existing approaches need power-hungry sensing hardware — always-on cameras drain the battery; wireless receivers use power-intensive analog components such as oscillators and high-speed ADCs. Second, they require rich signal processing capabilities — computing optical flows, FFTs, frequency-time Doppler profiles, etc., is computationally expensive; performing these operations while maintaining fast response times consumes power.

We introduce AllSee, a novel gesture-recognition system that can be used for computing devices, no matter how low-end and power-constrained they are. Such a capability can enable a number of interactive applications for Internet-of-things, where sensor deployments enable smart homes and cities [14, 24]; in fact, we show that AllSee can operate on battery-free devices that run off of harvested RF energy. AllSee can also be useful for small form-factor consumer electronics without the need for conventional input modalities like keypads. Finally, AllSee consumes three to four orders of magnitude lower power than existing systems; thus, it can also enable always-on gesture recognition on mobile devices such as

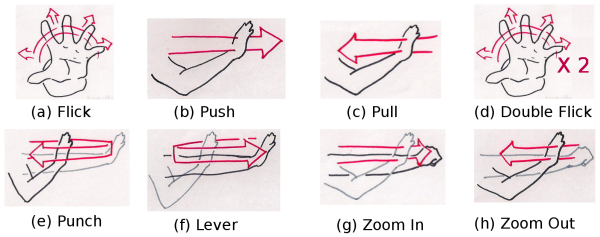

Figure 2: **Gesture Sketches.** AllSee can detect and classify these eight gestures using TV and RFID transmissions with average classification accuracies of 94.4% and 97% respectively.

smartphones and tablets.

AllSee's approach is to extract gesture information from ambient RF signals (e.g., TV transmissions) that already exist around us. AllSee can also work with signals from dedicated RF sources like RFID readers. AllSee eliminates the need for power-hungry wireless hardware components (e.g., oscillators) by using low-power analog operations to extract just the signal amplitude. To keep the computational complexity low, we avoid prior Doppler-based approaches that require computing FFTs and frequency-time Doppler profiles.[1] Instead, we introduce a novel design that extracts gesture information from the amplitude of the received signal.

At a high level, we use the insight that motion at a location farther from the receiver results in smaller wireless signal changes than from a close-by location. This is because the reflections from a farther location experience higher attenuation and hence have lower energy at the receiver. Now, consider the push and pull gestures shown in Fig. 2(b) and Fig. 2(c). As the user moves her arm *toward* the receiver, the wireless changes induced by the gesture increase with time, as the arm gets closer to the receiver. On the other hand, as the user moves her arm *away* from the receiver, the changes decrease with time. Thus, the receiver can distinguish between a pull and a push gesture even without access to the Doppler information. In §3, we explore this time-domain analysis further and extract deterministic rules to detect and classify the eight gestures in Fig. 2. Further in §3.3, we show that a subset of these rules can be encoded in the analog domain using passive components such as capacitors and resistors, further reducing our power consumption.

To demonstrate the feasibility of AllSee, we built multiple prototypes (one of which is shown in Fig. 1) that achieve gesture recognition on a range of devices. The first prototype is a modified RFID tag that extracts gesture information from signals of an RFID reader. The second prototype is a battery-free device that uses existing 725 MHz TV signals as a source of both power and gesture information. The third prototype encodes our gesture-detection rules using analog components, to further reduce the power consumption. We also integrate our prototypes with an off-the-shelf Nexus S phone and demonstrate gesture recognition in through-the-pocket scenarios; this enables the user to gesture at the phone in a pocket, to say change volume or mute the phone.

We evaluated our prototypes with up to five users using the gestures shown in Fig. 2. Our findings are as follows:

- AllSee classifies the eight gestures shown in Fig. 2 with an average accuracy of 97% and 94.4% on our RFID- and TV-based prototypes respectively; the accuracy is 92.5% with our phone prototype in through-the-pocket scenarios. This is promising, given that the accuracy for random guesses is 12.5%.
- AllSee achieves the above accuracies for distances of up to 2.5 feet from the device, while using 28.91 μW. This is in contrast to a state-of-the-art low-power gesture recognition system that consumes about 66 mW and has a range of upto 15 centimeters [31].
- The rate of false positive events—gesture detection in the absence of the target human—is 0.083 events per hour over a 24-hour period. AllSee achieves this by using a repetitive flick gesture to gain access to the system.
- AllSee's response time (the time between the gesture's end and its recognition) is less than 80 μs.
- Our analog gesture-encoding prototype that uses capacitors and resistors to detect a subset of gestures, consumes as little as 5.85 μW.

Contributions. We make the following contributions:

- We introduce the first gesture-recognition system that can run on battery-free devices. Our approach also enables always-on gesture recognition on mobile devices.
- We present computationally-light algorithms to extract gesture information from time-domain wireless signals.
- We show how to encode gestures in the analog domain using components such as capacitors and resistors.
- We build prototypes to demonstrate gesture recognition for multiple devices including RFID tags and power-harvesting battery-free devices. We integrate our prototypes with an off-the-shelf smartphone and use AllSee to classify gestures while the phone is in a pocket.

Our current implementation is limited to locations that have signals from either TV towers or RFID readers. We believe, however, that the techniques developed in this paper can be extended to leverage cellular and Wi-Fi transmissions, making AllSee more ubiquitous.

[1] As we show in §3.2.1, AllSee's receiver does not have phase information. Hence prior Doppler frequency solutions are in fact not applicable.

2 Related Work

Our work is related to prior art from both wireless and gesture-recognition systems.

(a) Wireless Systems: Prior work has shown the feasibility of using wireless signals for coarse motion detection (e.g., running [21] and walking forward and backward [18]). Further, our recent work on WiSee has shown how to extract gesture information from wireless signals [32]. These systems require power-hungry ultra-wideband transceivers [20, 33], interference-nulling hardware [18], or multiple antennas [18, 32]. Further, they require receivers with power-consuming analog components such as oscillators and high-speed ADCs and impose significant computational requirements such as 1024-point FFT and frequency-time Doppler profile computations [32]. We also note that these systems were implemented on USRPs, each of which consumes about 13.8 W [16]. In contrast, AllSee enables gesture recognition with orders of magnitude lower power.

AllSee is also related to work on low-power ultra-wideband radar sensors [1, 26, 27] that perform proximity detection. AllSee builds on this work, but goes beyond motion and velocity detection and designs the first wireless gesture-recognition system for power-constrained devices. We also show how to encode gestures using analog components with as little as 5.85 μW.

Finally, prior work [25, 29] has leveraged backscattered signals from RFID tags for activity recognition on a powered RFID reader. These systems, however, require quite a bit of computational processing and are designed to operate on powered RFID readers. In contrast, we enable gesture recognition on power-constrained devices.

(b) Gesture-Recognition Systems: Prior gesture-recognition systems can be primarily classified into vision-based, infrared-based, electric-field sensing, ultrasonic, and wearable approaches. Xbox Kinect [17], Leap Motion [9], PointGrab [12], and CrunchFish [5] use advances in cameras and computer vision to enable gesture recognition. Xbox Kinect uses the 3D sensor by PrimeSense which consumes 2.25 W [13], while PointGrab and CrunchFish run on mobile devices and consume as much power as the embedded camera.

Samsung Galaxy S4 introduced an "air gesture" feature that uses infrared cameras to enable gesture recognition. It is, however, not recommended to keep the gesture recognition system ON as it can drain the battery [2]. Further, it is known to be sensitive to lighting conditions [3] and does not work in through-the-pocket scenarios. GestIC [6], which we believe is the state-of-the-art system, uses electric-field sensing to enable gesture recognition using about 66 mW in the processing mode [31]. However, it requires the user's hand to be within 15 centimeters of the screen and also does not work in through-the-pocket scenarios. Further, it requires a relatively large surface area for sensing the electric fields. AllSee on the other hand achieves gesture recognition with three to four orders of magnitude lower power, works on devices with smaller form factors and in through-the-pocket scenarios.

Ultrasonic systems such as SoundWave [28] transmit ultrasound waves and analyze them for gesture recognition. These systems require active ultrasound transmissions and expensive operations such as 1024-point FFTs and Doppler profile computations. In contrast, AllSee leverages existing wireless signals (e.g., TV) and thus does not require active transmissions; this reduces the power consumption to the microwatts range. We note that, in principle, the time-domain analysis developed in this paper can be applied to ultrasonic systems to reduce their computational complexity.

Finally, prior work on inertial sensing and other on-body gesture recognition systems require instrumenting the human body with sensing devices [11, 22, 23]. In contrast, we focus on gesture recognition without requiring such instrumentation.

3 AllSee

AllSee is an ultra-low power wireless gesture-recognition sensor that consumes three to four orders of magnitude lower power than state-of-the-art systems. It uses ambient wireless signals (e.g., TV, cellular, and Wi-Fi) to extract gesture information. In this paper, we focus on demonstrating the feasibility of our designs using TV (and RFID) transmissions.

Designing such a wireless system is challenging for three main reasons: First, traditional radio receivers use power-intensive components such as oscillators that provide magnitude and phase information; the latter allows for gesture-recognition using Doppler frequency analysis. In contrast, AllSee uses passive components that significantly reduce the power consumption but provide only magnitude information. Thus, we need to develop algorithms and designs that can extract gesture information without relying on Doppler frequencies. Second, our designs should work on power-constrained devices and hence should be highly power-efficient and require minimal computational resources. Finally, gesture recognition is interactive in nature and hence requires short response times; this means that our algorithms and hardware should introduce minimal delays.

The rest of this section describes how AllSee addresses these challenges. We first describe AllSee's receiver design that extracts amplitude information using passive hardware components, and then present our algorithm to perform gesture classification using only this

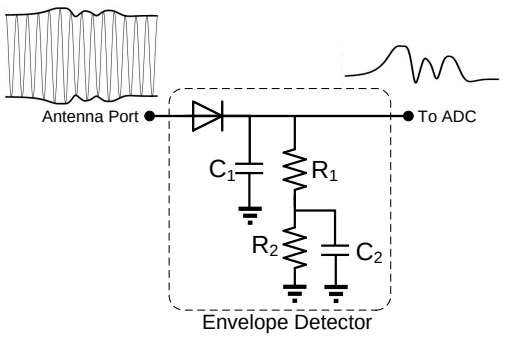

Figure 3: **AllSee's receiver circuit.** AllSee uses an envelope detector to extract the amplitude information. Using different values for the analog components, the receiver can work with both TV and RFID transmissions.

magnitude information.

3.1 AllSee's Receiver Design

We ask the following question: how can we extract amplitude information without using power-intensive components such as oscillators? Oscillators generate the carrier frequency (e.g., the 725 MHz TV carrier), and are typically used at receivers to remove the carrier frequency (down-conversion). At a high level, AllSee removes the carrier frequency by using an analog envelope detector. The rest of this section first describes how this works for constant-wave transmissions (e.g., RFID) and then extends it to work with fast-changing ambient TV transmissions.

(a) With Constant-Wave Transmissions: AllSee uses the envelope detector shown in Fig. 3 to remove the carrier frequency and extract amplitude information. As shown in the figure, the envelope detector tracks the envelope of the received signal, while eliminating the carrier frequency. The key point to note is that the envelope detector circuit is designed using passive analog components (diodes, resistors, and capacitors) and hence is ultra-low power in nature. The operating principle behind the design is similar to that used in RFID tags: To a first-order approximation, diodes act as switches that allow current to flow in the forward direction but not in the reverse; capacitors are charge-storage elements; and resistors regulate current flow. The diode in the above circuit provides charge to the capacitor C_1, whenever the input voltage is greater than the voltage at the capacitor. On the other hand, when the input is lower than the voltage at the capacitor, the diode does not provide charge and the resistors R_1 and R_2 slowly dissipate the energy stored on the capacitor, lowering its voltage. The rate of voltage drop is determined by the product $C_1 * (R_1 + R_2)$. Thus, by choosing appropriate values of R_1, R_2 and C_1, we can pick the rate at which the signal's envelope is tracked. Effectively, the above circuit acts as a low-pass filter, smoothing out the carrier in the constant-wave transmissions; the additional capacitor C_2 aids with this filtering.

Note that the envelope detector does not remove the amplitude variations caused by human gestures. This is because the circuit is tuned to track the variations caused by human motion which happen at a rate orders of magnitude lower than the carrier frequency.

(b) With Fast-Changing TV Transmissions: The key problem with ambient signals is that they have information encoded in them and hence have fast-varying amplitudes. For example, ATSC TV transmissions encode information using 8VSB modulation, which changes the instantaneous amplitude of the signal. In principle, the receiver could decode the TV transmissions and estimate the channel parameters to extract the amplitude changes that are due to human gestures. This is, however, infeasible on a power-constrained device. Thus, the challenge is to distinguish between the encoded TV information and the changes in the received signal due to human gestures, while operating on power-constrained devices.

Our key insight is that amplitude changes due to human gestures happen at a much lower rate than the changes inherent to TV transmissions. Our design leverages this difference in the rates to separate the two effects. Specifically, TV signals encode information at a rate of 6 MHz, but human gestures occur at a maximum rate of tens of Hertz. AllSee uses the envelope detector in Fig. 3 to distinguish between these rates. Specifically, we set the time constant of the envelope detector to be much greater than $\frac{1}{6MHz}$. This ensures that the encoded TV data is filtered out, leaving only the amplitude changes that are due to human gestures.

3.2 AllSee's Classification Logic

AllSee extracts gesture information from the signal output by the envelope detector. In this section, we first explain why prior approaches to wireless gesture recognition do not apply in our case. We then describe AllSee's algorithm to classify gestures.

3.2.1 Why Are Prior Approaches Not Applicable?

Prior approaches leverage wireless Doppler shifts for gesture classification. For example, the reflections from a user moving her hand toward the wireless receiver creates a positive Doppler shift. On the other hand, when the user moves her hand away from the receiver, it creates a negative Doppler shift. By using the sign of the Doppler shift, prior work [32] distinguishes between these gestures.

The problem is that the output of the envelope detector does not have phase information, which makes it difficult to apply the above approach. To understand this in more detail, let us consider a basic scenario where an RF source transmits a sinusoid with a frequency of f. Thus, the transmitted signal is given by:

$$sinftsinf_ct$$

where f_c is the transmitter's center frequency. Now, say that the user moves her arm towards the receiver and creates a Doppler shift, f_d. The receiver now receives the following signal [36]:

$$sinftsinf_ct + sinftsin(f_c + f_d)t$$

That is, the received signal is a linear combination of the direct signal from the RF source and the Doppler-shifted multi-path reflection from the user's arm. For simplicity, we assume that the user's reflection has the same signal strength as the direct signal, but the analysis would be similar in the general case. Now we can simplify the above equation to:

$$sinft(sinf_ct + sin(f_c + f_d)t) \quad (1)$$

$$= \quad 2sinftcos\tfrac{f_dt}{2}sin(f_c + \tfrac{f_d}{2})t \quad (2)$$

Traditional receivers use oscillators tuned to the center frequency f_c; hence they can extract the Doppler frequency f_d from the last sinusoid term in the above equation. AllSee, however, uses a low-power envelope-detector circuit that by nature is not as frequency-selective as an oscillator [19]. Specifically, the envelope detector tracks the envelope of the fastest-changing sinusoid in the received signal. Thus, in Eq. 2 the envelope detector considers the last sinusoid, $sin(f_c + \frac{f_d}{2})t$, as the effective transmitted signal and removes it. So, the output of the envelope detector is,

$$2sinftcos\tfrac{f_dt}{2}$$

$$= \quad sin(f + \tfrac{f_d}{2})t + sin(f - \tfrac{f_d}{2})t$$

Now if the receiver computes an FFT of the above signal, centered at f, it sees energy in both the positive and negative frequencies. This holds true independent of whether the user performs the push or the pull action. As a result, the receiver cannot distinguish between these two gestures using Doppler information.

Note that, from Eq. 2 the Doppler shift does not affect the transmitted signal $sinft$. Hence, replacing the transmitted sinusoid with any other signal does not change the above analysis.

3.2.2 AllSee's Time-Domain Analysis

AllSee leverages both the structure of the magnitude changes as well as the timing information to classify gestures. To see this, consider the push and pull gestures. As the user moves her arm towards the receiver, the changes in magnitude increase, as the arm gets closer to the receiver. This is because the reflections from the user's arm undergo lower attenuations as the arm gets closer. When the user moves her arm away from the receiver, the changes in the magnitude decrease with time. Thus, the changes in the time-domain signal can be uniquely mapped to the push and the pull gestures as shown in Figs. 4(b) and (c).

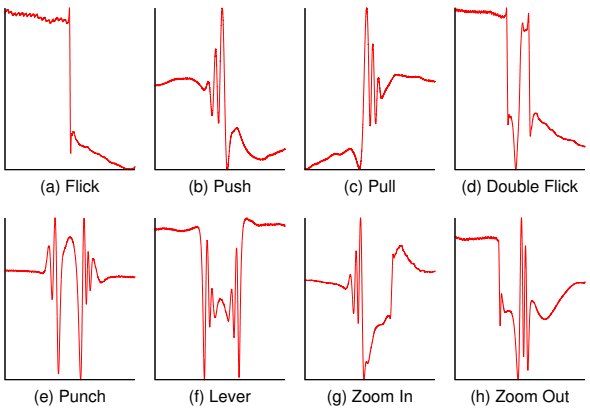

Figure 4: **Changes created on the envelope detector's output.** The amplitude changes have a unique correspondence to the gestures in Fig. 2.

AllSee also leverages timing information to classify gestures. Specifically, the wireless changes created by the flick gesture, as shown in Fig. 4(a), occurs for a shorter period of time compared to a push or a pull gesture. Using this timing information one can distinguish between these three gestures. Fig. 4 plots the amplitude changes created by our gestures as a function of time.

In the rest of this section, we describe in more detail how AllSee identifies and classifies these gestures. Specifically, AllSee uses a low-rate 10-bit ADC operating at 200 Hz to digitally process the signal output by the envelope detector. AllSee's time-domain classification algorithm has three main steps: (1) Signal conditioning to remove location dependence, (2) Segmentation to identify a time-domain segment that corresponds to a gesture, and (3) Classification to determine the most likely gesture amongst a set of gestures.

(1) Signal Conditioning: Our goal is to extract a deterministic location-independent mapping between gestures and amplitude changes. Note that the actual amplitudes vary with the user's position. For example, when the user performs the push gesture, the initial and final amplitudes depend on where the user starts and ends the push action. AllSee removes these location dependencies by performing a moving average over a time window (set to 320 ms in our implementation) and subtracting the average from

each digital sample returned by the ADC; thus, effectively normalizing the received signal.

(2) Segmentation: AllSee uses amplitude changes to detects the start and end of a gesture. Specifically, it computes a derivative of the received signal, i.e., the difference between the current and the previous sample. When this difference rises above a threshold (set to 17.5 mV in our implementation), AllSee detects the beginning of a gesture. Similarly when this difference falls below the same threshold, we detect the end of the gesture. The use of the derivative operation to detect the beginning and end of a gesture works because, as shown in Fig. 4, the changes caused by a gesture tend to be high. This results in large differences between adjacent samples, which we can use for segmentation.

We note the following: First, in comparison to ambient human motion such as walking and running, the changes between adjacent samples tend to be higher during intentional gestures close to the device. Thus, the above segmentation procedure helps reduce the false positive rate. Second, in some of our experiments, the difference between adjacent samples prematurely dropped below the threshold, even before the end of the gesture. It rises back up soon afterward, creating multiple close-by segments. To avoid this being detected as multiple gestures, we combine any two segments that occur within 75 milliseconds into a single segment.

(3) Gesture Classification: In principle, one could run signal-processing algorithms such as dynamic time warping to distinguish between the signals in Fig. 4. However, this is not desirable since it increases the computational complexity. Instead, AllSee uses simple rules that have minimal complexity to distinguish between gestures. For example, to classify between the push and pull gestures, we use the following rule: if the maximum changes in the signal occurs closer to the start of a gesture segment, it is a pull action; otherwise, it is a push action. In Alg. 1 we describe the rules for all eight of our gestures. We note that the algorithm is a set of if-then-else statements; in the worst case, these require only 56 instructions on an MSP430 microcontroller [10].

Algorithm 1 Gesture Classification

```
while Vsig − Vsig_prev < THRESHOLD do
    LOWPOWERSLEEP()
end while
[length, maxIndex] ← GETGESTURE()
g0 ← CLASSIFYSUBGESTURE(length, maxIndex)

[length, maxIndex] ← GETGESTURE()
g1 ← CLASSIFYSUBGESTURE(length, maxIndex)

if (g0 = FLICK and g1 = FLICK) then return D_FLICK
else if (g0 = FLICK and g1 = PULL) then return Z_OUT
else if (g0 = PUSH and g1 = FLICK) then return Z_IN
else if (g0 = PUSH and g1 = PULL) then return PUNCH
else if (g0 = PULL and g1 = PUSH) then return LEVER
else if (g0 = FLICK and g1 = NULL then return FLICK
else if (g0 = PUSH and g1 = NULL) then return PUSH
else if (g0 = PULL and g1 = NULL) then return PULL
end if

function CLASSIFYSUBGESTURE(length, maxIndex)
    if (length < FLICKLENGTH) then return FLICK
    else if (maxIndex < length/2) then return PUSH
    else if (maxIndex ≥ length/2) then return PULL
    end if
end function
```

3.3 Analog Gesture Decoding

In this section, we ask if we can further reduce the power consumption of the above design. As we see later in §5, the main factor that contribute to power consumption is the ADC. Specifically, we require an ADC with a resolution of eight to ten bits for the classification algorithm in §3.2 to work with high accuracy. So our goal is to eliminate the need for such high-resolution ADCs.

Our idea is to encode gesture information directly using analog components such as capacitors and resistors. Such an approach could reduce the need to processing the signals in the digital domain and hence avoids high-resolution ADCs. To show the feasibility of this idea, we design a circuit, shown in Fig. 5, that can distinguish between the punch and the flick gestures from Fig. 2. The circuit has four main components: an envelope detector to remove the carrier frequency, a second envelope detector that tracks time-domain changes caused by the gestures at a slow rate, an averaging circuit that computes the slow-moving average of the second envelope detector, and finally a low-power comparator that outputs bits.

Fig. 5 annotates the signals for the two gestures at each stage of the circuit. After the first envelope detector, the signals no longer have the carrier frequency. The second envelope detector tracks the signal at a much lower rate, and hence the punch signal looks like an increase and then a decrease in the amplitude levels; this corresponds to starting the arm at an initial state and then bringing it back to the same state. The flick signal, on the other hand, is a transition between two reflection states: one where the fingers are closed to another where the fingers are wide open.

The averaging circuit and the comparator allow us to distinguish between these two signals. Specifically, the averaging circuit further averages these signals to create the red signals shown in the figure. Now the comparator takes these signals and their average values as inputs, and outputs a '1' bit whenever the signal is greater than the

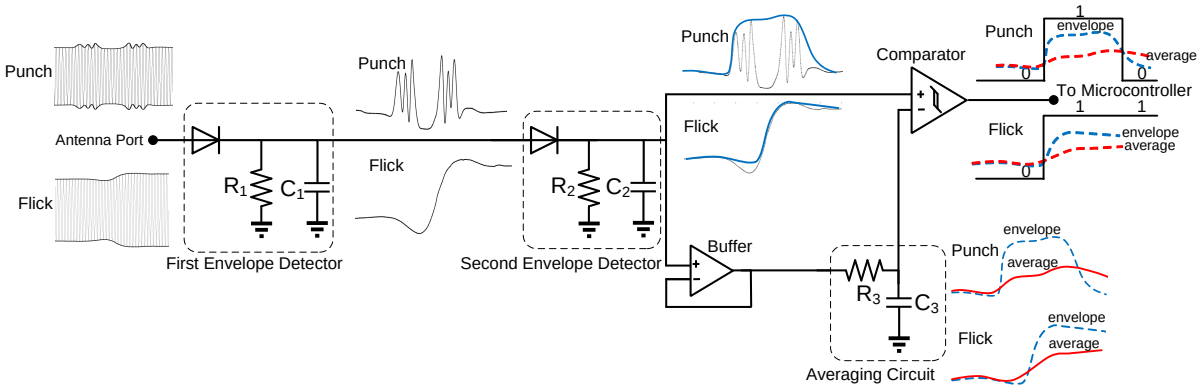

Figure 5: **AllSee's analog gesture encoding.** The circuit has four main components: an envelope detector to remove the carrier frequency, a second envelope detector that tracks changes caused by the gestures at a slow rate, an averaging circuit and finally a low-power comparator that outputs bits. The buffer ensures that the envelope detector and the averaging circuit do not affect each other. At each stage, input (dashed lines) and output (solid lines) waveforms corresponding to the punch and flick gestures are annotated.

average and a '0' bit otherwise. Thus, the comparator outputs unique set of bit patterns for the two gestures (010 and 011 in the figure). Thus, we can classify these gestures with almost no computational complexity.

We note three main points: First, the comparator is essentially a one-bit ADC; it has minimal resolution and hence consumes the least amount of power. Second, the parameters in our circuit are chosen to account for the timing information inherent in our specific gestures. Thus, it is unlikely that random human motions would trigger the same bit patterns. Third, while the above circuit only classifies the flick and the punch gestures, in principle, one can use higher order capacitor and resistor networks to encode all the eight gestures. This however is not in the scope of this paper.

4 Hardware Design

Fig. 6 shows the general hardware design of an AllSee device. It has two core components: an AllSee gesture receiver, and the computational logic that detects and classifies human gestures. Our implementation performs the digital logic operations on a low-power microcontroller. The AllSee gesture receiver primarily is the design in §3 that extracts amplitude information. However, it could also incorporate the analog gesture encoding mechanism described in §3.3.

The figure also shows optional components: a transmitter and receiver for communications, an RF energy harvester and the power management circuit to extract power from RF signals of either TV towers or RFID readers. These components are essential in devices such as RFID tags and ambient RF-powered devices, but are not necessary when AllSee is used in battery-powered mobile devices such as smartphones. We note that our design could, in principle, be incorporated into sensor hubs that are becoming popular on mobile devices.

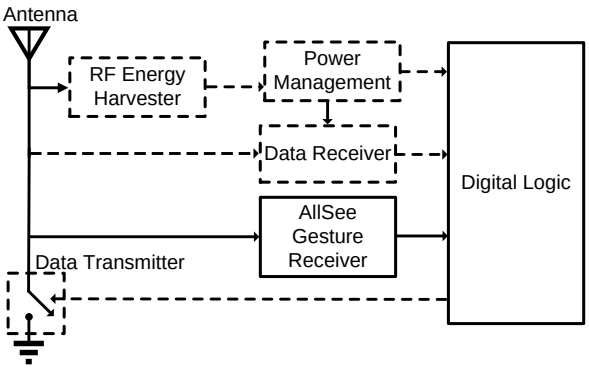

Figure 6: **AllSee's Hardware Design.** It consists of two main components: the wireless receiver for gesture recognition and our classification logic. The other components such as transmitter and receiver for communications, and energy harvester are optional.

5 Prototype Implementation

AllSee prototypes are implemented on two-layer printed circuit boards (PCBs) using off-the-shelf commercial circuit components. The PCBs were designed using the Altium design software and manufactured by Sunstone Circuits. The capacitor and resistor values R_1, R_2, C_1 and C_2 shown in Fig. 3 are set to 150 $k\Omega$, 10 $M\Omega$, 27 nF and 0.2 μF respectively. Our pluggable gesture recognition component consists of a low-power microcontroller (MSP430F5310 by Texas Instruments [10]) and an in-

Table 1: **AllSee's Average Power Consumption**

	ADC-based	Analog-based
No Gestures	26.96 μ W	4.57 μ W
15 Gestures/minute	28.91 μ W	5.85 μ W

terface to plug in our wireless receivers. It also features a UART interface to send data to a computer for debugging purposes as well as low-power LEDs. The output from the wireless receivers is sampled by an ADC at a frequency of 200 Hz (i.e., generating a digital sample every 5 ms). In most of our experiments, AllSee uses 10 bits of resolution at the ADC; in §6.1, however, we use other resolutions and sampling rates to understand their effects on classification accuracy.

To minimize power consumption, the microcontroller sleeps most of the time. The ADC wakes up the microcontroller to deliver digital samples every 5 ms. The microcontroller processes these samples before going back to sleep mode. The maximum time spent by the microcontroller processing a digital sample is 280 μs.

We also build a prototype for the analog gesture encoding system described in §3.3 by incorporating additional components into our wireless receivers. Specifically, we use an ultra-low power comparator, TS881 [15], and implement the buffer using an ultra-low power operational amplifier (ISL28194 [8]). The output of the comparator is fed to the digital input-output pin of the microcontroller. The capacitor and resistor values R_1, R_2, R_3, C_1, C_2 and C_3 shown in Fig. 5 are set to 470 $k\Omega$, 56 $M\Omega$, 20 $k\Omega$, 0.47 μF, 22 μF and 100 μF respectively. The RF energy harvester, transmitter, receiver and power-management circuit we use in our prototypes are similar to those in previous work [30, 34, 35].

Table 1 shows the total power consumption of our prototypes. For the ADC-based prototype, the 10-bit ADC continuously sampling at 200 Hz consumes 23.867 μW. The micro-controller consumes 3.09 μW for signal conditioning and gesture segmentation and 1.95 μW for gesture classification (as described in §3.2.2); the average power consumption is 26.96 μW when no gestures are present and 28.91 μW when classifying 15 gestures (including the starting gesture) per minute. In the analog-based prototype, the hardware components, the buffer and the comparator consume a total of 0.97 μW. The micro-controller consumes 3.6 μW in sleep mode (i.e., no bit transitions at the comparator's output). The average power consumption for the analog-based system is 4.57 μW when no gestures are present and 5.85 μW when classifying 15 gestures per minute.[2]

[2]Note that our prototypes use off-the shelf components and a general purpose micro-processor. One can further reduce the power consumption by using application specific integrated circuits.

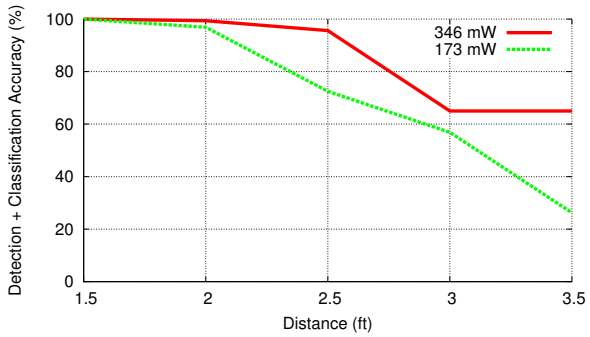

Figure 7: **Effect of distance and transmit power level.** The plots show results for two transmit power levels at the RFID reader. We operate in conservative settings – commercial RFID readers operate at up to 1 W.

6 Evaluation

We first present micro-benchmarks to understand the effect of various parameters on our system. We then evaluate different system aspects including classification accuracy, response time, and false-positive rate.

6.1 Micro-Benchmarks

We evaluate the key aspects that affect classification accuracy: (1) the user's distance from the prototype and the transmit power level, and (2) the bit-resolution and sampling rate of the ADC. Since it is easier to run controlled benchmark experiments with RFID readers than with TV towers, in this section we use our RFID prototype in the presence of an RFID reader.

Effect of distance and transmit power level: We run experiments to understand the effects of these parameters on classification accuracy. Specifically, we run our USRP-based RFID reader at two different transmit power levels: 346 mW and 173 mW. Note that commercial RFID readers can go upto 1 W; thus, we are operating in more conservative settings. For each of the power levels, we place our RFID-prototype in the decoding range of the reader. We then have the user stand at different distances from our prototype and perform our eight gestures, 20 times each, without fully blocking the signal from the RFID reader. Note that the users were not trained to orient themselves in a particular direction. At each of these distances, we compute the average classification accuracy across all eight gestures.

Fig. 7 shows classification accuracy as a function of the user's distance from our prototype. The plots show the following:

• As the distance between the user and the device increases, the classification accuracy decreases. This is expected since the strength of the wireless reflections from

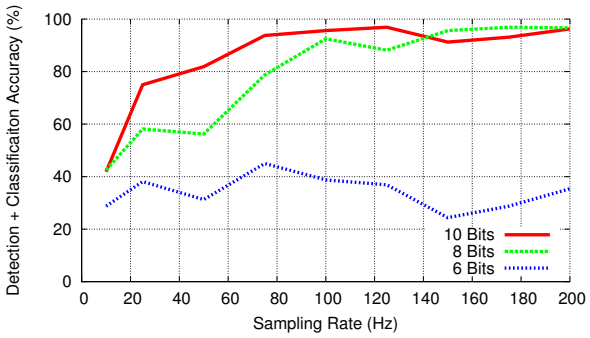

Figure 8: **Effect of ADC parameters.** The plot shows the accuracies at three different ADC bit resolutions.

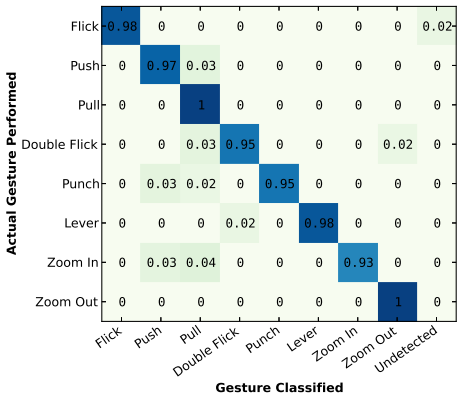

Figure 9: **Confusion matrix for our RFID prototype.** The average classification accuracy across all the gestures is 97%.

the user's body decreases as the user moves away from the device. This makes it harder to detect minute gestures such as the flick motion, bringing down the classification accuracy. We note, however, that the accuracies are greater than 90% at distances of 2.5 feet and 2 feet respectively for the two power levels. Such distances are sufficient for most gesture-recognition applications on mobile devices and sensors.

• As the transmit power level decreases, the classification accuracy decreases. This is because our wireless receiver has a minimum sensitivity below which it cannot accurately track changes from human gestures. We note, however, that the power levels we use in this experiment are lower than the 1 W power used by a commercial RFID reader. Further, our prototype RFID tags have about 11 dBm lower sensitivity than commercial tags. So while the trends we see here would hold for commercial tags, one can expect that the system would work at larger distances between the user and the prototype.

Effect of ADC parameters: The power consumption increases with the sampling rate and the bit resolution at the ADC. To empirically evaluate this effect, we run experiments in the presence of an RFID reader with a transmit power of 346 mW. We compute the classification accuracies for different ADC sampling frequencies and 6-bit, 8-bit and 10-bit resolutions. The user performs the eight gestures 20 times each, for each combination of sampling rates and resolutions.

Fig. 8 shows the detection and classification accuracies for the eight gestures shown in Fig. 2 as a function of the sampling rate. The different curves correspond to different bit resolutions. The figure shows that the accuracy typically increases with the ADC's bit resolution. The accuracy is also lower at low sampling rates; this is expected because as the sampling rates decrease, we lose timing information about the gestures.

6.2 AllSee's Classification Accuracy

We evaluate AllSee's accuracy in classifying gestures with our RFID-based and TV-based prototypes using the ADC design in §3.2.

Evaluating Our RFID-Based Prototype: We first evaluate the classification accuracy of gesture recognition using RFID signals.

Experiments: We run experiments in locations spanning two lab spaces in the UW CSE building. To check if AllSee works in the presence of multi-path reflections from nearby objects, one of these locations has strong reflective surfaces such as walls and objects (metallic cupboards, desks) close to our prototype hardware. In each of these locations, our prototype is placed in positions that are in the decoding range (about 1 meter) of an USRP-based RFID reader with a 346 mW transmit power. In our experiments, users perform the gestures in Fig. 2 at a distance of 1.5 to 2.5 feet away from the prototype. Before each experiment, users were shown how to perform all of the gestures. We ran the experiments with five users who volunteered to perform gestures; one of the five users is a co-author of this paper. Each gesture is performed a total of 20 times. The gestures are detected and classified on the microcontroller on our hardware prototype using the algorithm described in Alg. 1.

Results: Fig. 9 plots the confusion matrix where each row denotes the actual gesture performed by the user and each column the gestures it was classified into. The last column counts the fraction of gestures that were not detected at the receiver. Each element in the matrix corresponds to the fraction of gestures in the row that were classified as the gesture in the column; the fraction is computed across all the locations and the users. The table shows the following:

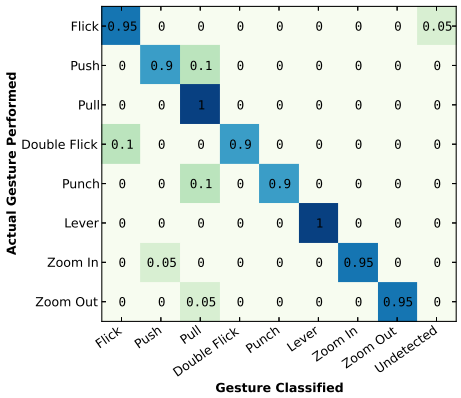

Figure 10: **Confusion matrix for our TV prototype.** The average classification accuracy across all the gestures is 94.4%. The lower accuracy is due to lower RF frequency of TV transmissions.

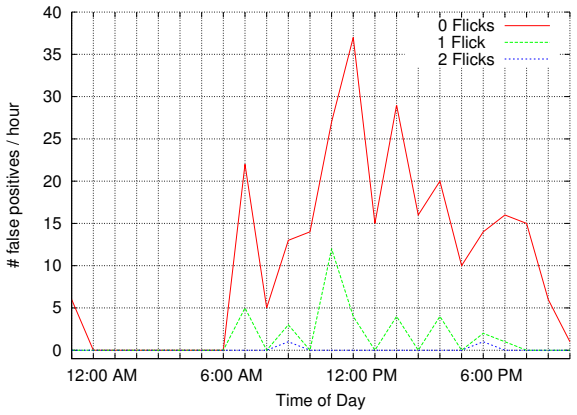

Figure 11: **False Positive Rate from a 24-Hour trace.** The figure plots the false positive rate when we use a flick gesture as a starting sequence.

• The average accuracy is 97% with a standard deviation of 2.51% when classifying among our eight gestures. This is in comparison to a random guess, which has an accuracy of 12.5% for eight gestures. This shows that one can extract rich information about gestures from time-domain wireless signals. We note that all the detection and classification operations were performed on our hardware prototype. This demonstrates the feasibility of achieving gesture recognition on battery-free devices.

• The flick motion was the most likely gesture to be undetected. This is because the flick gesture involves only finger movements; wireless reflections from the fingers have a much lower energy than those from the whole arm. Thus, the receiver has a higher probability of not detecting these gestures. Finally, there are small variations in the signals, due to differences in gesture articulation across users. These variation however are small and do not necessitate per-user tuning.

Evaluating Our TV-Based Prototype: Next we evaluate our accuracy using ambient TV signals.

Experiments: We run experiments using our battery-free prototype that harvests TV signals for power. We use the receiver design from our prior work on ambient backscatter devices [30] that can currently operate up to 10 Km away from a TV tower with power levels ranging between -24 dBm and -8 dBm. AllSee operates at similar distances and power levels from the TV tower. In principle, we can increase the distance and power sensitivity by using ASIC designs; this, however, is not in the scope of this paper. Our receiver prototype is tuned to harvest power and extract gesture information from TV signals in the 50 MHz band centered at 725 MHz. The users stand in a random location 1.5 to 2.5 feet away from the AllSee receiver and randomly perform all our eight gestures 20 times each. As above, the microcontroller on our prototype detects and classifies the gestures. We extract this information and compute the classification accuracy for the gestures.

Results: Fig. 10 shows the confusion matrix for our TV-based prototype. The figure shows similar trends to the results with our RFID-based prototype. The classification accuracies with our TV-based prototype, however, are lower than those with RFID signals. The main reason for this is the lower transmission frequency of TV signals. Specifically, lower transmission frequencies (higher wavelengths) require larger displacements to have a similar change in the received signals. Since RFID transmissions occur at 915 MHz, small displacements (e.g., a flick gesture) create large changes in the wireless signal. In contrast, since the TV transmissions occur at 725 MHz, the extent of wireless changes is smaller, making it harder to detect such gestures.

6.3 AllSee's False Positive Rate

To avoid random human motion near the device from being classified as the target gestures, AllSee uses a unique gesture sequence (a repetitive flick gesture) at the beginning to detect the target human. In this section, we evaluate the effectiveness of such an approach.

Experiments: Since our prototypes have the range of a few feet, we stress-test our system by placing them next to a participant's desk. Specifically, we run experiments in our lab over a 24-hour period during which the participant continues to perform activities including typing, eating, and moving around in the chair. Our prototype's location is such that five other lab occupants have to get as close as a foot to enter or leave their workspace.

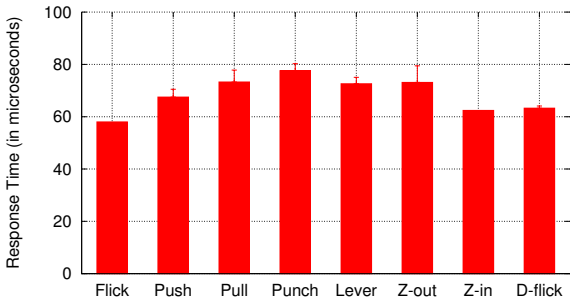

Figure 12: **Response time for various gestures.** The maximum response time is less than 80 μs across all the gestures. Z-in, Z-out, and D-flick refer to the zoom in, zoom out, and double flick gestures.

Results: Fig. 11 plots the average number of false detection events per hour as a function of time. The results show that when the receiver does not use a starting sequence (zero repetitions), the number of false positive events is about 11.1 per hour over the 24-hour period. We note that this is surprisingly low despite running experiments next to the user. This is because, as explained in §3.2, our segmentation algorithm is designed to only account for the large instantaneous changes in the received amplitude, that occur with intentional gestures; this reduces the probability of ambient motion resulting in the expected segment lengths. The results also show that the average number of false-positive events reduces to 1.46 per hour when a single flick action is used as a starting sequence. This is because the flick action creates a very short burst of amplitude changes that rarely occur with typical human activities. Note that as the number of flick repetitions increase, the false-positive rate reduces. Specifically, with two repetitions, this average rate reduces to 0.083 events per hour. This is again because a repetitive flick motion creates periodic short bursts of amplitude changes, which are unlikely to occur with typical activities. Further, since we expect these bursts to have a specific range of periodicities, random mechanical and environmental variations are unlikely to be confused for the repetitive flick sequence.

6.4 AllSee's Response Time

Short response times are important for the interactive nature of gesture recognition. Here, we evaluate AllSee's response time, i.e., the time between the completion of a gesture and its classification by our prototype. Recall that the microcontroller runs instructions at 1 MHz and the ADC operates at 200 Hz, waking up the microcontroller every 5 ms to deliver the digital samples. A 1 MHz microcontroller can run up to $10,000$ instructions in 10 ms. The number of instructions in Alg. 1 is significantly smaller and hence 10 ms is an upper bound on AllSee's response time.

Table 2: **Classification With Analog Gesture Encoding**

	Classification Rate
Punch Gesture	25/25
Flick Gesture	23/25

Experiments: To compute the exact response time, we measure the time difference between when the user finishes performing the gesture and when the microcontroller outputs the classified gesture. We program the microcontroller to toggle two output pins: first pin when it receives a data sample from the ADC and the second at the end of gesture classification. We observe the two output pins using an oscilloscope. In the absence of gestures, the first pin toggles periodically at 5 ms (sampling rate of ADC) and the second pin is steady. In the presence of a gesture, however, the second pin toggles immediately after classification. We compute the response time by measuring the time difference between the second pin's toggle and the periodic toggle on the first pin right before it. During the evaluation, the user performs our eight gestures 20 times and we compute the response time for each gesture averaged across the 20 repetitions.

Results: Fig. 12 shows the average measured response times for all the gestures. The plot shows the following:

- The maximum response time across all the gestures is less than 80 μs. This demonstrates that AllSee's gesture recognition algorithm requires negligible computation that can be performed even on an MSP430 with limited memory and computational capabilities.
- The variance of the response time, within the repetitions of the same gesture, is between 2–3 μs. This is because across experiments, the number of instructions that need to be run per gesture remains constant and deterministic. The only variability comes from the operational frequency of the microcontroller which is 1 MHz and hence has a resolution of 1 μs.

6.5 Evaluating Analog Gesture Encoding

Next, we evaluate our prototype for analog gesture encoding in the presence of RFID signals. The user stands at a distance of two feet away from our prototype tag and randomly performs the flick and punch gestures, 25 times each. Our prototype detects these gestures using analog components such as capacitors and resistors as described in §3.3. We extract the results and compute the classification rates for the two gestures.

Table 2 shows the classification results. They show that while the punch gesture was always correctly detected and classified, the flick gesture was misclassified

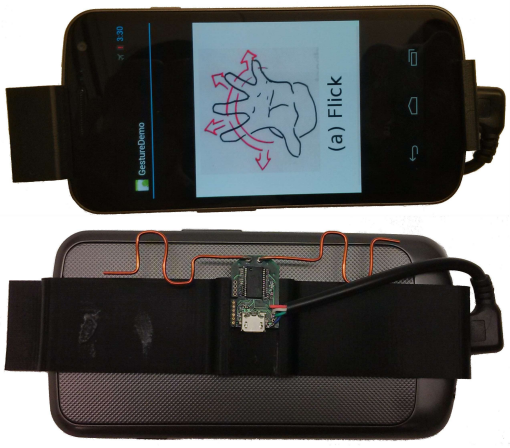

Figure 13: **Phone Prototype.** We interface a Galaxy Nexus with our miniaturized prototype mounted on a 3D printed phone case. We achieve 92.5% accuracy for the gestures in Fig. 2, in through-the-pocket scenarios.

in 2 out of the 25 trials. This is because the flick action involves finger motion that has less effect on the wireless signals than the arm motion in the punch action. Note that the average accuracy across the two gestures is about 96%. These results show the feasibility of our analog gesture encoding approach.

6.6 Through-the-Pocket Gestures

The ability to change the music or mute the phone while it is in a pocket, purse, or jacket is a useful capability. In this section, we integrate our miniaturized hardware prototype with a Samsung Galaxy Nexus smartphone [7] and demonstrate gesture recognition in such scenarios. We designed a 3-D printed case to mount our prototype on the phone. Further, as shown in Fig. 13, we modified the antenna for a better form factor and connect our prototype to the phone via a USB OTG cable.

We evaluate our smartphone prototype by placing the device in the pocket of a jacket that the user is wearing. The user then performs the eight gestures in Fig. 2 on the same x-y plane as the phone, 20 times each. Our results show that the mean accuracy across gestures is about 92.5%, which is encouraging. We note two main points: First, in comparison to the previous scenarios, the classification accuracy here is a bit lower. This is because, in these experiments, our device is hidden behind the jacket fabric and hence experiences higher signal attenuation. Second, our proof-of-concept prototype is limited to scenarios where the user is stationary and does not walk/run while performing the gestures. In principle, one can leverage other low-power sensors such as accelerometers on the phone to detect these scenarios; this however is not in the scope of this paper.

7 Discussion

We describe how one may augment AllSee's current design to make it more ubiquitous and robust.

Leveraging cellular and Wi-Fi signals: Our current prototypes work with TV and RFID transmissions. However, building AllSee prototypes that work with other RF signals such as Wi-Fi and cellular transmissions can enable greater ubiquity. We believe that the techniques developed in this paper can provide the framework for such designs. For instance, one could design specific envelope detectors that focus on human gestures while eliminating uncorrelated changes caused by the burstiness in these systems. Further, to address the issue of non-continuous transmissions, one can leverage the periodically transmitted beacons or pilot symbols in Wi-Fi and cellular systems to extract gesture information.

Reducing the gesture recognition range: One of the advantages of using wireless signals is that they enable far-field gesture recognition. However, one could reduce the sensitivity of our wireless receivers to reduce their range. This could be beneficial in certain applications where proximity is used as a proxy for access control.

Integration with other power sources: Our current design uses existing signals (e.g., TV and RFID transmissions) as a source of both power as well as gesture information. In principle, however, one can use AllSee to enable gesture recognition on other harvesting devices where we extract gesture information from wireless signals but use solar or mechanical energy for harvesting.

8 Conclusion

We introduce AllSee, a novel gesture recognition system that consumes three to four orders of magnitude lower power than the state-of-the-art systems today. AllSee can operate on battery-free devices such as ambient RF powered devices and RFID tags; it also enables always-on gesture recognition on mobile devices such as smartphones and tablets. We build prototypes and demonstrate that our system can detect and classify a set of eight gestures with classification accuracies as high as 97%. We believe that this is a promising result and hope that the techniques developed in this paper would take us closer to the vision of ubiquitous gesture interaction.

Acknowledgements: We thank the members of the UW Networks and Wireless group, David Wetherall, Ben Ransford, our shepherd Lili Qiu, and the anonymous NSDI reviewers for their helpful comments and Lilian de Greef for help with the gesture sketches in Fig. 2. This research is funded in part by a Google Faculty Research Award and a Washington Research Foundation gift.

References

[1] Advantaca. http://www.advantaca.com/AboutAdvantaca.htm.

[2] Air Gesture on Samsung S4 drains battery. http://www.1techportal.com/2013/05/maximize-your-galaxy-s4s-battery-in-five-ways/.

[3] Air Gesture on Samsung S4 works well under very well lit conditions. http://touchlessgeneration.com/discover-touchless/testing-of-air-gestures-on-the-galaxy-s4/#.UkSVWIY3uSo.

[4] AN580: Ifrared gesture recognition by Silicon Labs. http://www.silabs.com/Support%20Documents/TechnicalDocs/AN580.pdf.

[5] CrunchFish. http://crunchfish.com/.

[6] GestIC: electrical near field (E-field) 3D Tracking and Gesture Controller by Microchip. http://www.microchip.com/pagehandler/en-us/technology/gestic/home.html.

[7] Google Nexus S. http://www.android.com/devices/detail/nexus-s.

[8] ISL28194 op amp datasheet by Intersil. http://www.intersil.com/content/dam/Intersil/documents/fn62/fn6236.pdf.

[9] Leap Motion. https://www.leapmotion.com/.

[10] MSP430F5310 micro-controller by Texas Instruments. http://www.ti.com/product/msp430f5310.

[11] Myo by Thalmic Labs. https://www.thalmic.com/en/myo/.

[12] Point Grab. http://www.pointgrab.com/.

[13] The PrimeSense 3D Awareness Sensor. http://www.primesense.com/wp-content/uploads/2012/12/PrimeSense-3D-Sensor-A4-Lo.pdf.

[14] Smart Things. http://www.smartthings.com/.

[15] TS 881 comparator datasheet by STMicroelectronics. http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/DM00057901.pdf.

[16] USRPN200/N210 by Ettus Research. https://www.ettus.com/content/files/07495_Ettus_N200-210_DS_Flyer_HR.pdf.

[17] Xbox Kinect. http://www.xbox.com/en-US/kinect.

[18] F. Adib and D. Katabi. Seeing Through Walls Using WiFi! In *SIGCOMM*, 2013.

[19] R. Barnett, G. Balachandran, S. Lazar, B. Kramer, G. Konnail, S. Rajasekhar, and V. Drobny. A passive uhf rfid transponder for epc gen 2 with -14dbm sensitivity in 0.13 μm cmos. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pages 582–623, 2007.

[20] G. Charvat, L. Kempel, E. Rothwell, C. Coleman, and E. Mokole. A Through-dielectric Radar Imaging System. In *Trans. Antennas and Propagation, 2010*.

[21] K. Chetty, G. Smith, and K. Woodbridge. Through-the-wall Sensing of Personnel Using Passive Bistatic WiFi Radar at Standoff Distances. In *Trans. Geoscience and Remote Sensing, 2012*.

[22] G. Cohn, S. Gupta, T.-J. Lee, D. Morris, J. R. Smith, M. S. Reynolds, D. S. Tan, and S. N. Patel. An ultra-low-power human body motion sensor using static electric field sensing. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12. ACM, 2012.

[23] G. Cohn, D. Morris, S. Patel, and D. Tan. Humantenna: using the body as an antenna for real-time whole-body interaction. In *CHI'12*.

[24] G. Cohn, E. Stuntebeck, J. Pandey, B. Otis, G. D. Abowd, and S. N. Patel. Snupi: sensor nodes utilizing powerline infrastructure. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10. ACM, 2010.

[25] A. Czeskis, K. Koscher, J. R. Smith, and T. Kohno. Rfids and secret handshakes: defending against ghost-and-leech attacks and unauthorized reads with context-aware communications. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08. ACM, 2008.

[26] P. Dutta, A. Arora, and S. Bibyk. Towards radar-enabled sensor networks. In *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, pages 467–474, 2006.

[27] P. Dutta and A. K. Arora. Integrating micropower radar and motes. IEEE Conf. on Ultra Wideband Systems and Technologies, Baltimore, MD, 2002.

[28] S. Gupta, D. Morris, S. Patel, and D. Tan. Soundwave: using the doppler effect to sense gestures. In *HCI 2012*.

[29] L. Kriara, M. Alsup, G. Corbellini, M. Trotter, J. D. Griffin, and S. Mangold. RFID Shakables: Pairing Radio-Frequency Identification Tags with the Help of Gesture Recognition. In *ACM CoNEXT*, 2013.

[30] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient Backscatter: Wireless Communication Out of Thin Air. In *ACM SIGCOMM*, 2013.

[31] Microchip. GestIC: Single-Zone 3D Tracking and Gesture Controller Data Sheet, 2012-2013.

[32] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. Whole-Home Gesture Recognition Using Wireless Signals. In *MOBICOM*, 2013.

[33] T. Ralston, G. Charvat, and J. Peabody. Real-time through-wall imaging using an ultrawideband MIMO phased array radar system. In *Array, 2010*.

[34] A. Sample and J. R. Smith. Experimental results with two wireless power transfer systems. In *Radio and Wireless Symposium, 2009. RWS '09. IEEE*, pages 16 –18, jan. 2009.

[35] A. Sample, D. Yeager, P. Powledge, A. Mamishev, and J. Smith. Design of an rfid-based battery-free programmable sensing platform. *IEEE Transactions on Instrumentation and Measurement*, 57(11):2608–2615, November 2008.

[36] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge University Press, New York, NY, USA, 2005.