WiGest: A Ubiquitous WiFi-based Gesture Recognition System

Heba Abdelnasser
Computer and Sys. Eng. Department
Alexandria University
heba.abdelnasser@alexu.edu.eg

Moustafa Youssef
Wireless Research Center
Egypt-Japan Univ. of Sc. and Tech.
moustafa.youssef@ejust.edu.eg

Khaled A. Harras
Computer Science Department
Carnegie Mellon University
kharras@cs.cmu.edu

***Abstract*—We present WiGest: a system that leverages changes in WiFi signal strength to sense in-air hand gestures around the user's mobile device. Compared to related work, WiGest is unique in using standard WiFi equipment, with no modifications, and no training for gesture recognition. The system identifies different signal change primitives, from which we construct mutually independent gesture families. These families can be mapped to distinguishable application actions. We address various challenges including cleaning the noisy signals, gesture type and attributes detection, reducing false positives due to interfering humans, and adapting to changing signal polarity. We implement a proof-of-concept prototype using off-the-shelf laptops and extensively evaluate the system in both an office environment and a typical apartment with standard WiFi access points. Our results show that *WiGest* detects the basic primitives with an accuracy of 87.5% using a single AP only, including through-the-wall non-line-of-sight scenarios. This accuracy increases to 96% using three overheard APs. In addition, when evaluating the system using a multi-media player application, we achieve a classification accuracy of 96%. This accuracy is robust to the presence of other interfering humans, highlighting *WiGest*'s ability to enable future ubiquitous hands-free gesture-based interaction with mobile devices.**

I. Introduction

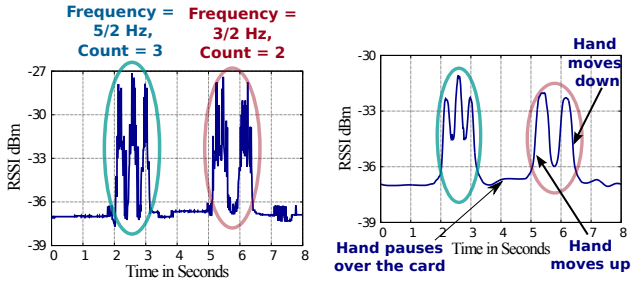
The exponential growth in mobile technologies has reignited the investigation of novel human-computer interfaces (HCI) through which users can control various applications. Motivated by freeing the user from specialized devices and leveraging natural and contextually relevant human movements, gesture recognition systems are becoming increasingly popular as a fundamental approach for providing HCI alternatives. Indeed, there is a rising trend in the adoption of gesture recognition systems into various consumer electronics and mobile devices, including smartphones [1], laptops [2], navigation devices [3], and gaming consoles [4]. These systems, along with research enhancing them by exploiting the wide range of sensors available on such devices, generally adopt various techniques for recognizing gestures including computer vision [4], inertial sensors [5]–[7], ultra-sonic [1], and infrared (e.g. on the Samsung S4 phone). While promising, these techniques experience various limitations such as being tailored for specific applications, sensitivity to lighting, high installation and instrumentation overhead, requiring holding the mobile device, and/or requiring additional sensors to be worn or installed.

With the ubiquity of WiFi-enabled devices and infrastructure, WiFi-based gesture recognition systems, e.g. [8]–[10], have recently been proposed to help overcome the above limitations in addition to enabling users to provide ***in-air hands-free*** input to various applications running on mobile devices. This is particulary useful in cases when a user's hands are wet, dirty, busy, or she is wearing gloves; rendering touch input difficult. These WiFi-based systems are based on analyzing the changes in the characteristics of the wireless signals, such as the received signal strength indicator (RSSI) or detailed channel state information (CSI), caused by human motion. However, due to the noisy wireless channel and complex wireless propagation, these systems either require **calibration** of the area of interest or **major changes** to the standard WiFi hardware to extract the desired signal features, therefore limiting the adoption of these solutions using off-the-shelf components. Moreover, all these systems do not provide **fine-grained** control of a specific user mobile device.

This paper presents *WiGest*, a ubiquitous WiFi-based hand gesture recognition system for controlling applications running on off-the-shelf WiFi-equipped devices. *WiGest* does not require additional sensors, is resilient to changes within the environment, does not require training, and can operate in non-line-of-sight scenarios. The basic idea is to leverage the effect of the in-air hand motion on the wireless signal strength received by the device from an access point to recognize the performed gesture. Figure 1 demonstrates the impact of some hand motion gestures within proximity of the receiver on the RSSI values, creating three unique signal states (we call primitives): a rising edge, a falling edge, and a pause. *WiGest* parses combinations of these primitives along with other parameters, such as the speed and magnitude of each primitive, to detect various gestures. These gestures, in turn, can be mapped to distinguishable application actions. Since *WiGest* works with a specific user device (e.g. a laptop or mobile phone), it has the advantage of removing the ambiguity of the user location relative to the receiver. This helps eliminate the requirement of calibration and special hardware.

There are several challenges, however, that need to be addressed to realize *WiGest*. These challenges include handling the noisy RSSI values due to multipath interference, medium contention, and other electromagnetic noise in the wireless medium; handling the variations of gestures and their attributes for different humans and even for the same human at different times; handling interference (i.e. avoiding false positives) due

978-1-4799-8381-0/15/$31.00 ©2015 IEEE



(a) Raw signal (b) Denoised signal

Fig. 1: The impact of two hand motion frequencies on the RSSI. The received signal is composed of three primitives: rising edge, falling edge, and pause. The five motion repetitions can also be counted.

to the motion of other people within proximity of the user's device; and finally be energy-efficient to suit mobile devices.

To address these challenges, *WiGest* leverages different signal processing techniques that can preserve signal details while filtering out the noise and variations in the signal. In addition, we leverage multiple overheard APs as well as introduce a unique signal pattern (acting as a preamble) to identify the beginning of the gesture engagement phase; this helps counter the effect of interfering humans, increase robustness, and enhance accuracy. Finally, *WiGest* energy-efficiency stems from the fact that detecting the preamble can be efficiently done based on a simple thresholding approach, rendering the system idle most of the time. In addition, we use an efficient implementation of the wavelet transform that has a linear time complexity in various signal processing modules.

We implement *WiGest* on off-the-shelf laptops and evaluate its performance with three different users in a 1300ft^2 three bedroom apartment and a 5600ft^2 floor within our engineering building. Various realistic scenarios are tested covering more than 1000 primitive actions and gestures each in the presence of interfering users in the same room as well as other people moving in the same floor during their daily life.

Our results reveal that *WiGest* can detect the basic primitives with an accuracy of 87.5% using a single AP only for distances up to 26 *ft* including through-the-wall non-line-of-sight scenarios. This accuracy increases to 96% using three overheard APs, which is the typical case for many WiFi deployment scenarios. In addition, when evaluating the system using a multi-media player application case study, we achieve a classification accuracy of 96%. This accuracy is robust to the presence of other interfering humans.

The rest of the paper is organized as follows. Section II provides an overview on related work. Sections III and IV present an overview and the details of the *WiGest* system respectively. We evaluate the system in Section V. Finally, we conclude the paper and provide directions for future work in Section VI.

II. Related Work

Gesture recognition systems generally adopt various techniques such as computer vision [4], inertial sensors [5], ultrasonic [1], and infrared electromagnetic radiation (e.g. on Samsung S4). While promising, these techniques suffer from limitations such as sensitivity to lighting, high installation and instrumentation overhead, demanding dedicated sensors to be worn or installed, or requiring line-of-sight communication between the user and the sensor. These limitations promoted exploiting WiFi, already installed on most user devices and abundant within infrastructure, for *activity* and *gesture* recognition as detailed in this section.

A. WiFi-based Activity Recognition Systems

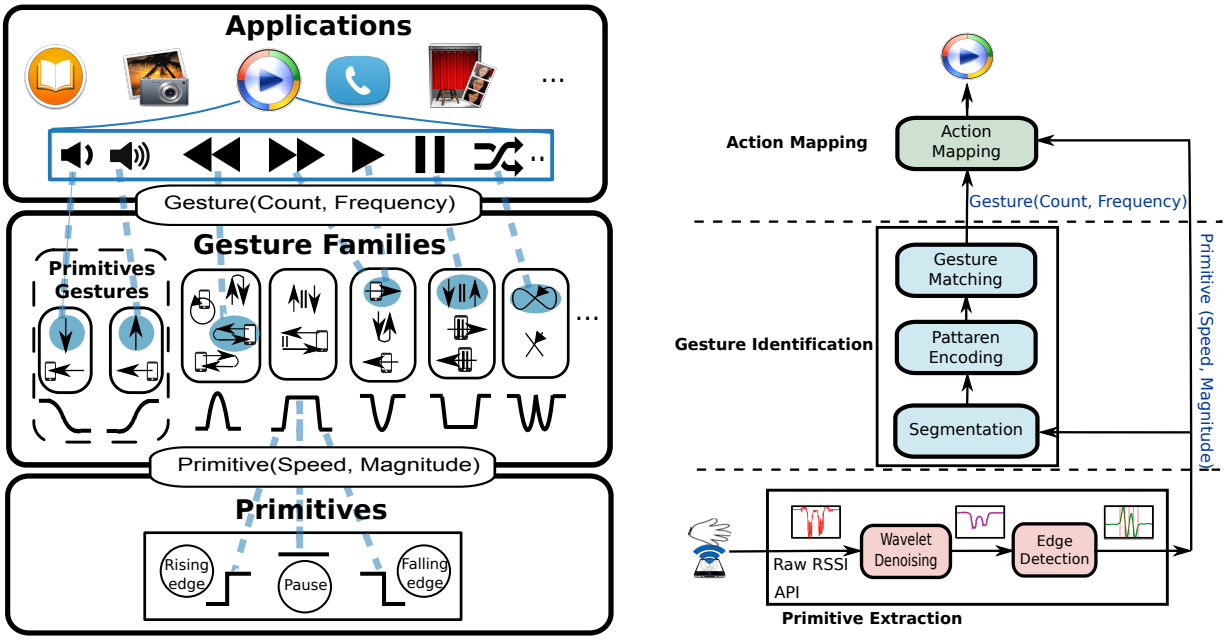
Device-free activity recognition relying on RSSI fluctuations, or the more detailed channel state information (CSI) in standard WiFi networks, have emerged as a ubiquitous solution for presence detection [11]–[22], tracking [23]–[28], and recognition of human activities [29]–[32]. For activity recognition systems, RFTraffic [29] introduces a system that classifies the traffic density scenarios based on the emitted RF-noise from the vehicles, while [30], [32] can further differentiate between vehicles and humans and detect the vehicle speed. In [33], authors provide localization and detect basic human activities such as standing and walking, using ambient FM broadcast signals, or by sensing WiFi RSSI changes within proximity of a mobile device. In addition, activities conducted by multiple individuals can be detected simultaneously with good accuracy utilizing multiple receivers and simple features and classifier systems [31], which can be further used in novel application domains, such as automatic indoor semantic identification [34]–[36].

The solutions described above typically require some prior form of training, have been tested in controlled environments, or work in scenarios where human presence is rare (e.g. intrusion detection), and most importantly, do not detect fine-grained motions such as hand gestures near mobile devices. *WiGest* builds on this foundational work to achieve fine-grained hand gesture recognition based on RSSI changes **without requiring** any training and in typical daily scenarios.

B. RF-based Gesture Recognition Systems

Work in this area represents the most recent efforts to detect fine-grained mobility and human gestures by leveraging RF signals. WiVi [9] uses an inverse synthetic aperture radar (ISAR) technique by treating the motion of a human body as an antenna array to track the resulting RF beam, thus enabling radar-like vision and simple through-wall gesture-based communication. Finally, WiSee [10] is a fine-grained gesture recognition system that builds on DopLink [37] and [38] by exploiting the doppler shift in narrow bands extracted form wide-band OFDM transmissions to recognize nine different human gesturers.

While these solutions provide high accuracy, they all require **special hardware** in order to employ their solutions. *WiGest*, on the other hand, works with off-the-shelf WiFi components, making it significantly more deployable. Moreover, working with off-the-shelf components requires dealing with more challenges in terms of noise and interference sources.



(a) Layers overview. (b) System logic flow.

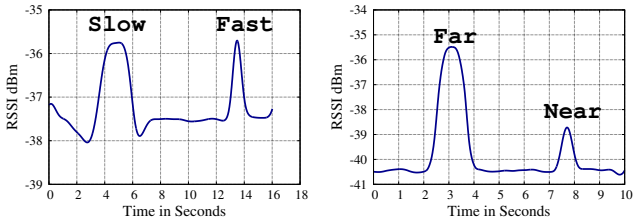
Fig. 2: *WiGest* (a) conceptual view and (b) processing components. *WiGest* has three layers: primitives, gestures, and applications. The figure shows a typical example for mapping the application actions to different example gestures and primitives. We note that multiple application actions (e.g. "play" and "fast" forward) can be mapped to the same gesture with different parameters (e.g., one count for play and more than one count for fast forward).

III. WiGest Conceptual Overview

In this section, we provide a conceptual overview of *WiGest* covering the Primitives, Gesture Families, and Application Actions layers (Figure 2a).

A. Primitives Layer

This layer detects the basic changes in the raw RSSI. These changes include rising edges caused by moving the hand away from the device, falling edges caused by moving the hand towards the device, and pauses caused by holding the hand still over the device. Higher layer gestures can be composed by combining these three primitives. There are other parameters that can be associated with these primitive actions. In particular, for the rising and falling edges, *WiGest* can extract the speed of motion and magnitude of signal change (Figure 3). To reduce the noise effect, *WiGest* discretizes the values of the parameters. Therefore, there are three defined speeds (high, medium, and low) and two defined magnitudes (far and near).



(a) Hand motion speed effect: The slower the speed, the slower the RSSI change. (b) Hand motion distance effect. The higher the distance, the higher the change in RSSI.

Fig. 3: Parameters associated with raw signal primitives.

(a) Right-left (b) Up-down (c) Infinity (d) Open-close

Fig. 4: Some gesture sketches detected by *WiGest*.

B. Gesture Families Layer

Different basic primitives from the lower layer can be combined to produce higher level gestures (Figure 4). For example, an up-down hand gesture can be mapped to the primitives falling then rising edges. Since there may be ambiguity between different hand gestures, we define the concept of a *gesture family*, which represents a set of gestures that have the same sequence of primitives. For example, as shown in Figure 2a, all up-down, right-left and left-right hand gestures have the same effect on the signal strength and hence the same primitives sequence of a falling then rising edge. Gesture families give flexibility to the developers to choose the best gesture that fits their applications from a certain gesture family. Similar to the basic primitives, gestures can have associated attributes. Specifically, the count of the gesture repetition (e.g. how many consecutive up-down gestures) and the frequency (how fast the repetition is performed) are two attributes that can be associated with the gestures (Figure 1b).

C. Application Actions Layer

This layer is where each application maps its actions to different gestures. Typically, one action is mapped to one gesture family and the developer can pick one or more gestures from the same family to represent an action. As an example, for a media player application (Figure 2) a "play" action can be performed with a right-movement hand gesture while a "volume up" action can be mapped to moving the hand up. The hand movement speed can be mapped to the rate of volume change.

In the next section, we give the details of extracting these different semantics and the associated challenges.

IV. The WiGest System

In this section, we discuss the processing flow of our system depicted in Figure 2b and address the associated challenges. This flow includes the three main stages that map to the different semantic layers: Primitives Extraction, Gesture Identification, and Action Mapping.

A. Primitives Extraction

The goal of this stage is to extract the basic primitives (rising edges, falling edges, and pauses) from the raw signal. Due to the noisy wireless signal, this stage starts by a noise reduction step using the Discrete Wavelet Transform (DWT), followed by an edge extraction and primitives detection step. We start with a brief background on the DWT. Then we give the details of the different submodules.

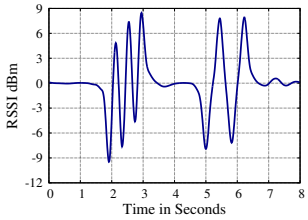
1) Discrete Wavelet Transform: Wavelet Transform provides a time-frequency representation of a signal. It has two main advantages: (a) an optimal resolution both in the time and the frequency domains; and (b) it achieves fine-grained multi-scale analysis [39]. In Discrete Wavelet Transform (DWT), the generic step splits the signal into two parts: an approximation coefficient vector and a detail coefficient vector. This splitting is applied recursively a number of steps (i.e. levels), J, to the approximation coefficients vector only to obtain finer details from the signal. At the end, DWT produces a coarse approximation projection (scaling) coefficients $\alpha^{(J)}$, together with a sequence of finer detail projection (wavelet) coefficients $\beta^{(1)}, \beta^{(2)}, ..., \beta^{(J)}$. The DWT coefficients in each level can be computed using the following equations:

$$\alpha_k^{(J)} = \langle x_n, g^{(J)}_{n-2^J k} \rangle_n = \sum_{n \in \mathbb{Z}} x_n \; g^{(J)}_{n-2^J k}, \qquad J \in \mathbb{Z} \tag{1}$$

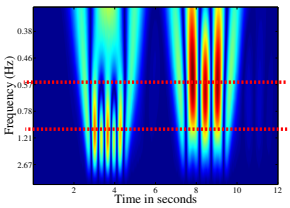
$$\beta_k^{(\ell)} = \langle x_n, h^{(\ell)}_{n-2^\ell k} \rangle_n = \sum_{n \in \mathbb{Z}} x_n \; h^{(\ell)}_{n-2^\ell k}, \qquad \ell \in \{1, 2, ..., J\} \tag{2}$$

where x_n is the n^{th} input point, $\langle . \rangle$ is the dot product operation, and g's and h's are two sets of discrete orthogonal functions called the wavelet basis (we used the Haar basis functions in our system). The inverse DWT is given by

$$x_n = \sum_{k \in \mathbb{Z}} \alpha_k^{(J)} \; g^{(J)}_{n-2^J k} + \sum_{\ell=1}^{J} \sum_{k \in \mathbb{Z}} \beta_k^{(\ell)} \; h^{(\ell)}_{n-2^\ell k} \tag{3}$$



(a) Detail coefficients of a denoised signal containing two motion speeds



(b) Spectrogram

Fig. 5: Detecting frequencies of interest dynamically using wavelet transform. The dashed lines in the spectrogram map to the two frequencies of interest.

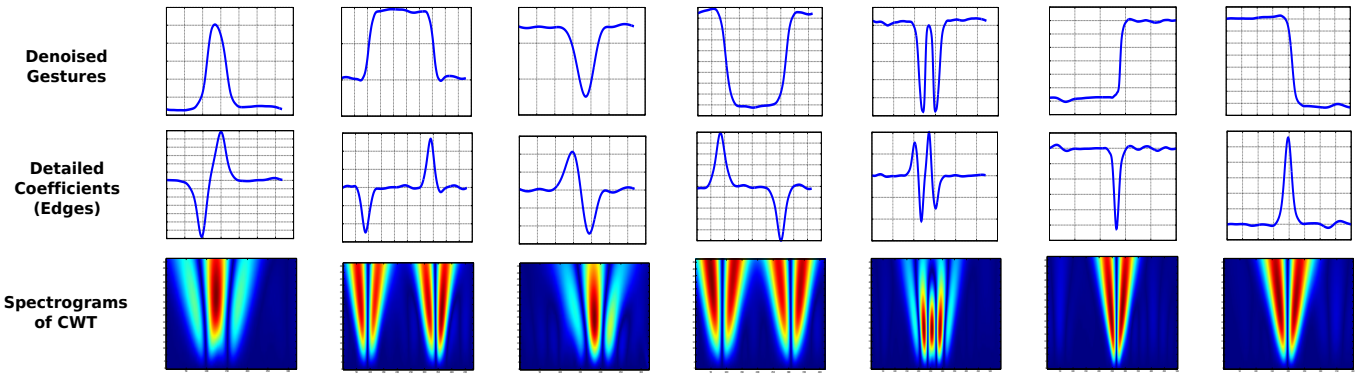
2) Noise Reduction: Due to complex wireless propagation and interaction with surrounding objects, RSSI values at the receiver are noisy. This noise can add false edges and affect *WiGest* accuracy and robustness. Therefore, to increase *WiGest* quality, we leverage a wavelet-based denoising method [40]. Wavelet denoising consists of three stages: decomposition, thresholding detail coefficients, and reconstruction.

In *decomposition*, DWT is recursively applied to break the signal into high-frequency coefficients (details) and low-frequency coefficients (approximations) at different frequency levels. *Thresholding* is then applied to the wavelet detail coefficients to remove their noisy part. The threshold is chosen dynamically, based on minimizing the Stein unbiased risk estimate (SURE), under the assumption of Gaussian noise [40]. Finally, *reconstruction* of the denoised signal occurs by combining the coefficients of the last approximation level with all thresholded details.

Wavelet denosing has the advantage of being computationally efficient (linear time complexity) to fit on mobile devices. In addition, it does not make any particular assumptions about the nature of the signal, and permits discontinuities in the signal [41]. Moreover, we leverage DWT in other system functionalities, allowing code sharing for better efficiency.

3) Edge Extraction and Primitives Detection: The main effect of the human hand motions on the received signal (detection primitives) are either rising edges, falling edges, or pauses. The place of these primitives need to be defined in time. However, since the human hand motion speed may be different from one person to another or for the same person at different times, we need a technique that can provide variable frequency resolution. Therefore, we use wavelet analysis for edge detection in *WiGest*. Compared to other techniques, e.g. the Fourier transform, Wavelet analysis provides both time and frequency locality at different resolutions as well as being computationally efficient.

Figure 5 shows the spectrogram of a denoised signal containing two different hand motion velocities. The figure shows that Wavelet transform can capture/localize the two frequencies of motion (the two dashed lines in the figure) using simple 2D local maximum extraction technique. This technique is then utilized in the DWT process to determine the levels of interest that are used in the edge detection stage.



(a) Up-Down (b) Up-||-Down (c) Down-Up (d) Down-||-Up (e) Right-Left (f) Up (g) Down

Fig. 6: Different hand gestures, their detected edges, and spectrograms. The first row is the denoised RSSI signals. The second row shows the fifth level details coefficients of the denoising module used in edge detection. The third row shows the spectrogram of the signal only used to determine the frequency of interest level (not for distinguishing gestures.)

Based on the DWT theory, a rising edge causes a local minima peak and a falling edge causes a local maxima peak in the detailed coefficients as shown in the second row of Figure 6. A pause is detected as a stable RSSI value (RSSI variance within a threshold) for a certain minimum time (0.5 second in our system). Therefore, using this module, any input signal can be translated into a set of raising edges, falling edges, and/or pauses. Figure 6 shows some examples of different hand motion gestures and the detected edge positions as well as their spectrograms. The figure also shows that the edge locations and frequency can be accurately extracted.

We note that *WiGest* can use multiple overheard APs to increase the system accuracy and resilience to noise. In this case, a majority vote on the detected primitive is used to fuse the detection of the different APs.

4) Primitives attributes: After extracting edges, each edge can be labeled with two parameters: speed and magnitude. Based on the edge duration, there are three possible speeds: high (edge duration < 0.75 sec), medium (0.75 sec $<$ edge duration < 1.5 sec), or low (edge duration > 1.5 sec). On the other hand, the distance value is also discretized into two values, high (more than 0.55 ft above the receiver) and low (less than 0.55 ft above the receiver), which is determined based on a threshold applied to the RSSI. Since the amplitude of the signal change is relative to the original signal level (i.e. a strong signal will lead to a lager change with the user hand movement and vice versa), we base the threshold value on the change of the start of the signal as we describe in the signal segmentation section below.

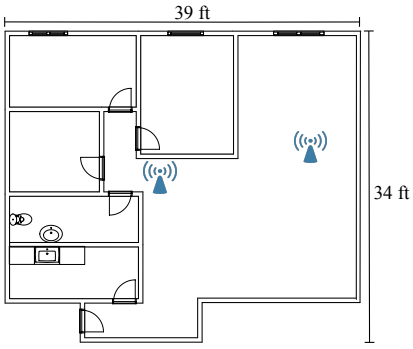
B. Gesture Identification

The goal of this processing stage is to extract the different gestures and their attributes (frequency and count). This is performed through two steps: segmentation and identification/matching.

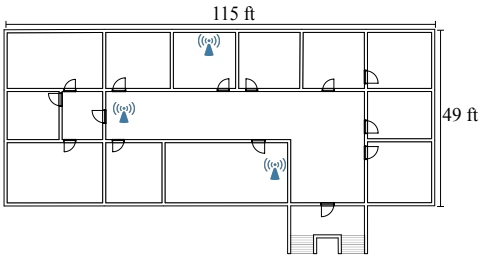
1) Segmentation: One important question for the correct operation of *WiGest* is when to know that the user is generating a gesture. This helps avoid false gestures generated by other actions/noise in the environment from nearby users and also leads to energy-efficiency. *WiGest* answers this question by using a special preamble that is hard to confuse with other actions in the environment. Without this preamble, the system refrains from interpreting gestures. The preamble contains two states: a drop in the input signal values and then two up-down signals. This is equivalent to holding the hand over the receiver and then making two up-down gestures. The first stage is detected by a simple and efficient thresholding operation. So in real-time, the default case is that the preamble extraction module only searches for a drop in the RSSI values. If detected, it moves to the second stage and searches for two consecutive up-down motions, which is equivalent to searching for four consecutive peaks in the detailed components of the DWT (computed in linear time). Once the preamble is detected, the communication channel begins between the device and the user, and the system scans for various gestures based on the primitives extracted in the previous section. Gesture recognition is terminated after a silence period preset by the user, returning to the preamble scanning phase.

In some cases, due to the complex wireless propagation environment, flipping between rising and falling edges may occur. To compensate for this, *WiGest* leverages the preamble. Specifically, the direction of the change of the first state (typically expected to be a drop in signal strength) determines whether the signal should be flipped or not. *WiGest* also leverages the preamble to determine the threshold magnitude (high/low) and motion frequency. Essentially, the extracted features from the predefined preamble help the system adapt to different users and different environments.

2) Pattern Encoding and Matching/Gesture Identification: Once the gesture boundary is determined and primitives are extracted, we convert the primitives to a string sequence: rising edges to positive signs, falling edges to negative signs, and pauses to zeros. The extracted string pattern is then compared with gesture templates to find the best match, as well as extract the count and frequency attributes (i.e. number of gesture repetitions per unit time).



(a) First test environment (apartment).



(b) Second test environment (one floor of the engineering building where our lab is located). The APs shown are the ones installed in this floor.

Fig. 7: Floorplans of the two test environments used for evaluating *WiGest*.

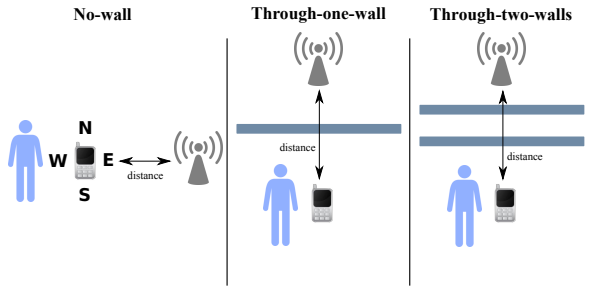


Fig. 8: The three scenarios layouts: the orientations characters in no-wall scenario indicate the possible four user orientations around her device relative to the AP.

C. Action Mapping

This is a direct mapping step based on the application semantics. The developer determines which application actions are mapped to which gesture families. In addition, the attributes of the gesture are passed to the application for further processing. For example, the frequency attribute can be used to determine how fast the character should run in a game, the count parameter can determine how far we should go in the list of choices, etc. Note that multiple actions can be mapped to the same gesture family if they have multiple attributes. For example, in a media player application, "play" can be mapped to a right-hand gesture, while fast forward can be mapped to a double right-hand gesture.

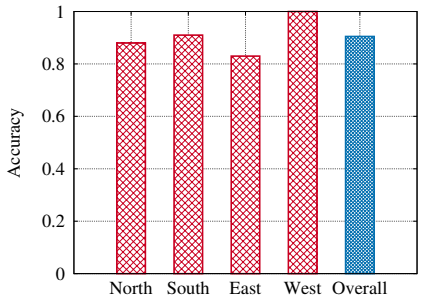


Fig. 9: Effect of the user location relative to the LOS between the device and the transmitter. The orientations are indicated in Figure 8.

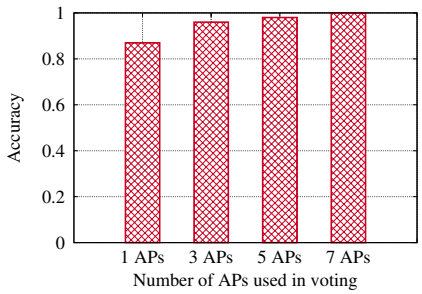


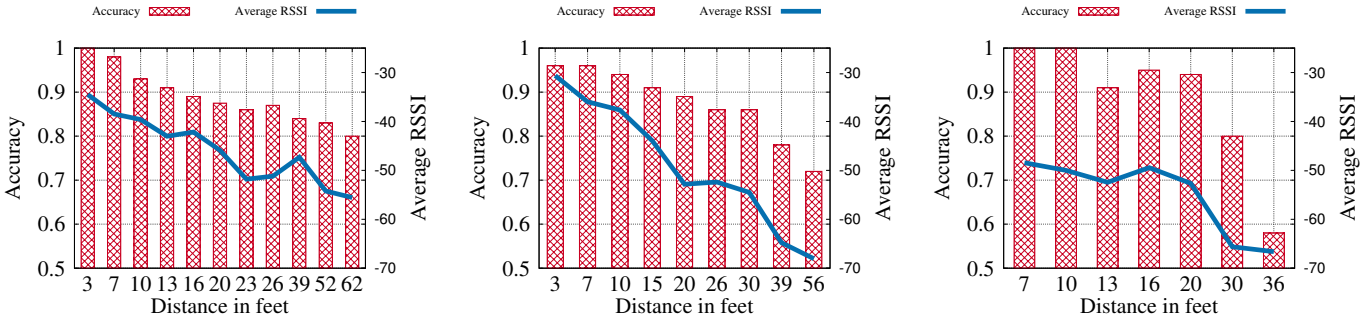
Fig. 10: Effect of increasing the number of APs on the system accuracy.

V. Evaluation

In this section, we analyze the performance of *WiGest* in two typical environments: (a) A typical apartment (Figure 7a) covering a 34×39 ft^2 area and consisting of a living room connected to the dining area, three bedrooms, kitchen, and a bathroom. The walls have a thickness of 0.5ft and the doors are made of wood and have a thickness of 0.16ft. (b) The second floor of an engineering building at our campus (Figure 7b) which contains 12 rooms connected by a large corridor and covers an area of 115×49 ft^2.

In both environments, an HP EliteBook laptop is used as a receiver with a sampling rate of 50 Hz controlled by the user using *WiGest*. The apartment had two Cisco Linksys X2000 APs while the engineering building has three Netgear N300 APs installed in the experiment floor. We experimented with three different users producing more than 1000 primitive actions and hundreds of gestures including scenarios with seven interfering users in the same room, in addition to other people moving in the same floor during their daily life.

In addition to evaluating *WiGest* at different locations uniformly distributed over the two environments, we extensively evaluate it in three different scenarios shown in Figure 8: **1) No-wall/Line-of-sight**: The laptop and AP are in the same room. **2) Through-one-wall**: The laptop and AP are placed in adjacent rooms separated by one wall. **3) Through-two-walls**: The laptop and AP are placed in two different rooms separated by a corridor. Thus the signal penetrates two walls.



(a) No-wall (b) Through-one-wall (c) Through-two-walls

Fig. 11: Impact of distance on edge detection accuracy. The figure also shows the average RSSI value at each distance.

(a) Primitives (b) Edge magnitude (c) Different speeds of motion

Fig. 12: Primitives Layer evaluation. The magnitude parameter is discretized to two values: low (distance less than 0.5ft and high for distance above 0.6ft above the mobile device. We also discretize the speed to three values: high (edge duration < 0.75 sec), medium (0.75 sec $<$ edge duration between < 1.5 sec), or low (edge duration > 1.5 sec).

We first evaluate the accuracy of detecting primitives and gestures under these three scenarios with different realistic settings. Then we evaluate *WiGest*'s performance at different locations uniformly distributed over both environments. Unless otherwise specified, all results are averaged over both test environments.

A. Primitives Layer Evaluation

1) Impact of Orientation: We first evaluate edge detection performance at four orientations of the user relative to the device (East, West, North, and South) in the no-wall scenario (Figure 8). The distance between the mobile device and the AP is set to 14ft. The results in Figure 9 show that the overall primitive detection accuracy of *WiGest* averaged over all different orientations is 90.5%. The highest accuracy is achieved in the West orientation while the lowest is achieved in the East orientation. This is intuitive because in the East orientation, the human body blocks the line-of-sight between the device and the AP causing two negative effects: First, it produces noise in the received signal. Second, it reduces the received RSSI due to attenuation through the human body; stronger RSSI leads to better accuracy as we quantify later. We note that accuracy can be further enhanced by combining the RSSI from multiple APs as we show in the next section.

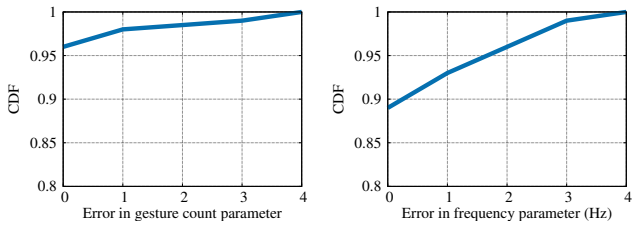
2) Impact of Multiple APs: Figure 10 shows the effect of using more than one AP on primitives detection accuracy averaged over different orientations. This experiment is performed in the engineering building due to the abundance of APs there. The overheard APs are over different floors in the building. A majority vote between all APs is used to select the correct primitive. The figure shows that increasing the number of APs to three increases the system accuracy to 96%, and reaches 100% when the device hears seven APs. Given the dense deployment of WiFi APs, this highlights that *WiGest* can achieve robust performance in the many cases.

3) Impact of Distance: We investigate the relation between *WiGest*'s accuracy and the coverage distance of an AP for the scenarios in Figure 8. In this experiment, the distance between the device and AP varies and each primitive is repeated 25 times at each location for different orientations. Figure 11 shows that the overall primitives detection accuracy decreases when increasing the distance between the device and AP. This is because stronger RSSIs lead to higher SNR and hence better accuracy. In other words, a weaker signal leads to lower changes in the signal strength in response to the hand movement, leading to less sensitivity. *WiGest*, however, can still achieve more than 87% accuracy for distances up to 26ft. This can be further enhanced by combining RSSIs from different APs as shown in the previous section.

4) Individual primitives and attributes detection accuracy: Finally, we evaluate the true positive (1- false negative) and false positive detection rate for the three primitives in addition to correctly identifying the primitive's speed and distance parameters. The distance between the mobile device and the AP is set to 14ft. The collected dataset contains 350 samples

	Down-Up	Inf.	Up-Down	Up-Pause-Down	Down-Pause-Up	Un-known
Down-Up	**0.94**	0	0	0	0	0.06
Infinity	0.07	**0.83**	0.04	0	0.03	0.03
Up-Down	0	0	**1**	0	0	0
Up-Pause-Down	0	0	0.05	**0.9**	0	0.05
Down-Pause-Up	0	0	0	0	**0.96**	0.04

TABLE I: Confusion matrix for the different gesture families.



(a) CDF of error in count parameter.

(b) CDF of error in frequency parameter.

Fig. 13: Gesture families parameters detection evaluation.

of each primitive for a total of 1050 samples. Figure 12 shows the results. The figure shows that the different *WiGest* processing modules lead to high detection accuracy with a high true positive rate and low false positive rate concurrently. In addition, *WiGest* can detect the primitives' attributes with high accuracy, allowing it to be used in different applications.

B. Gesture Families Layer Evaluation

We now evaluate the detection accuracy of the gesture families and their associated parameters (count and frequency). We consider the seven gesture families in Figure 2a.

1) Gesture detection: A user performs a total of 192 gestures at a distance of 14ft from the AP for the four different orientations relative to the mobile device using the AP the mobile device is associated with. The resulting confusion matrix is shown in Table I. Overall, these results show that *WiGest* can detect the gestures with a high accuracy of 92.6%. This accuracy is higher than the primitives detection accuracy due to the processing we apply on gestures as described in Section IV-B2. This can be further increased by leveraging more APs.

2) Gesture attributes detection accuracy: Figure 13 shows the accuracy of detecting various gesture parameters. The figure shows that *WiGest* can accurately detect the exact count 96% of the time. This percentage increases to 98% for a count error of one. Similarly, it can detect the repetition frequency of a gesture within one second 93% of the time.

C. Whole-home Gesture Recognition Case Study

We evaluate the system in our apartment environment using a multi-media player application. Both APs installed in the apartment are used, and since there is no majority vote here, the AP with the strongest signal is used. We evaluate the multi-media player action detection performance at eight locations uniformly distributed over the apartment and covering different

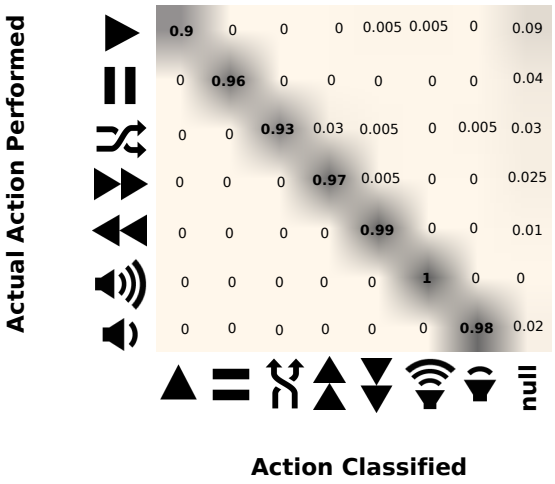


TABLE II: Confusion matrix for the media player application actions.

scenarios (line-of-sight, through-one-wall, and through-two-walls). At each location, the user performs the seven actions shown in Figure 2a. Each action is performed 20 times at each location for a total of 1120 actions. Table II shows the confusion matrix for the seven gestures across all locations. The matrix shows that the overall classification accuracy over all application actions is 96%. This accuracy shows *WiGest*'s ability to use wireless signals to extract rich gestures.

VI. Conclusion

We presented *WiGest*, a robust gesture recognition system that uses WiFi RSSI's to detect human hand motions around a user device. *WiGest* does not require any modification to the available wireless equipment or any extra sensors, and does not require any training prior to deployment. We addressed system challenges including signal denoising, gesture primitives extraction, reducing the false positive rate of gesture detection, and adapting to changing signal polarity caused by environmental interference. *WiGest*'s energy-efficiency stems from using a preamble that is easy to detect based on a simple thresholding approach as well as using the wavelet transform that works in linear time in its various processing modules.

Extensive evaluation of *WiGest* in two environments shows that it can detect basic primitives with an accuracy of 87.5% using a single AP for distances up to 26 *ft* including through-the-wall non-line-of-sight scenarios. This accuracy increases to 96% using three overheard APs, which is the typical case for many urban WiFi scenarios. In addition, *WiGest* can achieve a classification accuracy of 96% for the application actions. These results are robust to the presence of other interfering humans, highlighting *WiGest*'s ability for enabling future ubiquitous hands-free gesture-based interaction with mobile devices.

Currently, we are extending the system in multiple directions including leveraging detailed channel state information (CSI) from the physical layer to further enhance accuracy and generate finer grained gesture families, leveraging other ubiq-

uitous wireless technologies, such as cellular and bluetooth, among others.

REFERENCES

[1] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1911–1914.

[2] R. Block, "Toshiba Qosmio G55 features SpursEngine, visual gesture controls," http://www.engadget.com/2008/06/14/toshiba-qosmio-g55-features-spursengine-visual-gesture-controls/, last accessed March 1st, 2014.

[3] A. Santos, "Pioneer's latest Raku Navi GPS units take commands from hand gestures," http://www.engadget.com/2012/10/07/pioneer-raku-navi-gps-hand-gesture-controlled/, last accessed March 1st, 2014.

[4] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[5] G. Cohn, D. Morris, S. Patel, and D. Tan, "Humantenna: using the body as an antenna for real-time whole-body interaction," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1901–1910.

[6] C. Harrison, D. Tan, and D. Morris, "Skinput: appropriating the body as an input surface," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 453–462.

[7] D. Kim, O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis, and P. Olivier, "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*. ACM, 2012, pp. 167–176.

[8] M. Scholz, S. Sigg, H. R. Schmidtke, and M. Beigl, "Challenges for device-free radio-based activity recognition," in *Workshop on Context Systems, Design, Evaluation and Optimisation*, 2011.

[9] F. Adib and D. Katabi, "See through walls with wifi!" in *Proceedings of the ACM SIGCOMM*. ACM, 2013, pp. 75–86.

[10] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 27–38.

[11] A. E. Kosba, A. Saeed, and M. Youssef, "Rasid: A robust WLAN device-free passive motion detection system," in *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*. IEEE, 2012, pp. 180–189.

[12] M. Youssef, M. Mah, and A. Agrawala, "Challenges: device-free passive localization for wireless environments," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM, 2007, pp. 222–229.

[13] A. Saeed, A. E. Kosba, and M. Youssef, "Ichnaea: A low-overhead robust WLAN device-free passive localization system," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 1, pp. 5–15, 2014.

[14] H. Abdel-Nasser, R. Samir, I. Sabek, and M. Youssef, "MonoPHY: Mono-stream-based device-free WLAN localization via physical layer information," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013, pp. 4546–4551.

[15] K. El-Kafrawy, M. Youssef, and A. El-Keyi, "Impact of the human motion on the variance of the received signal strength of wireless links," in *PIMRC*, 2011, pp. 1208–1212.

[16] H. Aly and M. Youssef, "New insights into wifi-based device-free localization," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 541–548.

[17] M. A. Seifeldin, A. F. El-keyi, and M. A. Youssef, "Kalman filter-based tracking of a device-free passive entity in wireless environments," in *ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, 2011.

[18] M. Seifeldin and M. Youssef, "A deterministic large-scale device-free passive localization system for wireless environments," in *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2010, p. 51.

[19] A. Eleryan, M. Elsabagh, and M. Youssef, "Synthetic generation of radio maps for device-free passive localization," in *IEEE Global Telecommunications Conference (GLOBECOM 2011)*.

[20] A. E. Kosba, A. Saeed, and M. Youssef, "Robust WLAN device-free passive motion detection," in *IEEE Wireless Communications and Networking Conference (WCNC 2012)*, pp. 3284–3289.

[21] I. Sabek and M. Youssef, "Multi-entity device-free WLAN localization," in *IEEE GLOBECOM 2012*.

[22] M. Moussa and M. Youssef, "Smart devices for smart environments: Device-free passive detection in real environments," in *IEEE Percom Workshops 2009.*, 2009.

[23] J. Xiao, K. Wu, Y. Yi, L. Wang, and L. M. Ni, "Pilot: Passive device-free indoor localization using channel state information," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*. IEEE, 2013, pp. 236–245.

[24] A. E. Kosba, A. Abdelkader, and M. Youssef, "Analysis of a device-free passive tracking system in typical wireless environments," in *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*. IEEE, 2009, pp. 1–5.

[25] K. El-Kafrawy, M. Youssef, A. El-Keyi, and A. Naguib, "Propagation modeling for accurate indoor WLAN RSS-based localization," in *IEEE Vehicular Technology Conference Fall (VTC 2010-Fall)*.

[26] M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, and M. Youssef, "Nuzzer: A large-scale device-free passive localization system for wireless environments," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 7, pp. 1321–1334, 2013.

[27] I. Sabek and M. Youssef, "Spot: An accurate and efficient multi-entity device-free WLAN localization system," *CoRR*, vol. abs/1207.4265, 2012.

[28] I. Sabek, M. Youssef, and A. Vasilakos, "ACE: An accurate and efficient multi-entity device-free WLAN localization system," *IEEE Transactions on Mobile Computing*, 2014.

[29] Y. Ding, B. Banitalebi, T. Miyaki, and M. Beigl, "RFTraffic: passive traffic awareness based on emitted RF noise from the vehicles," in *IEEE 2011 ITS Telecommunications Conference*.

[30] A. Al-Husseiny and M. Youssef, "RF-based traffic detection and identification," in *IEEE Vehicular Technology Conference (VTC Fall), 2012*, 2012.

[31] S. Sigg, S. Shi, and Y. Ji, "RF-based device-free recognition of simultaneously conducted activities," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 531–540.

[32] N. Kassem, A. E. Kosba, and M. Youssef, "RF-based vehicle detection and speed estimation," in *IEEE VTC Spring*, 2012.

[33] S. Shi, S. Sigg, and Y. Ji, "Joint localization and activity recognition from ambient fm broadcast signals," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 521–530.

[34] M. Alzantot and M. Youssef, "CrowdInside: automatic construction of indoor floorplans," in *ACM 2012 SIGSpatial GIS Conference*.

[35] M. ELhamshary and M. Youssef, "CheckInside: An indoor location-based social network." in *ACM Ubicomp 2014*.

[36] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *ACM MobiSys 2012*.

[37] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, and S. Patel, "Doplink: using the doppler effect for multi-device interaction," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 2013, pp. 583–586.

[38] Y. Kim and H. Ling, "Human activity classification based on micro-doppler signatures using a support vector machine," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 47, no. 5, pp. 1328–1337, 2009.

[39] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed. Academic Press, 2008.

[40] S. Sardy, P. Tseng, and A. Bruce, "Robust wavelet denoising," *Signal Processing, IEEE Transactions on*, vol. 49, no. 6, pp. 1146–1152, 2001.

[41] S. Nibhanupudi, "Signal denoising using wavelets," Master's thesis, University of Cincinnati, 2003.