Unobtrusive Occupancy Detection with FastGRNN on Resource-Constrained BLE Devices

Md Fazlay Rabbi Masum Billah
University of Virginia
masum@virginia.edu

Bradford Campbell
University of Virginia
bradjc@virginia.edu

ABSTRACT

The emerging area of device-free occupancy detection (DfOD) has seen slow adoption due to deployability, scalability, and energy efficiency concerns resulting from the use of large, costly, and power-hungry devices like laptops and Wi-Fi routers in the state-of-the-art solutions. Moreover, these approaches often rely on cloud-offloading for data processing which requires extra communication latency and energy. To overcome these challenges, we develop an RF-based DfOD system using easily-deployable Bluetooth Low Energy (BLE) devices. Our system uses a kilobyte-sized machine learning algorithm running on the BLE device to predict the occupancy of a room from a small number of wireless packets, thereby enabling energy-frugal real-time analytics. We validate our approach with experiments in two indoor rooms using four nRF52840 BLE radios. Initial results suggest our system can detect occupancy of an indoor environment with 95% accuracy, 96% precision, and 92% recall while drawing a meager amount of current.

CCS CONCEPTS

• **Computer systems organization → Embedded systems, Sensor networks**;

KEYWORDS

BLE, RNN, FastGRNN, Occupancy detection

ACM Reference format:
Md Fazlay Rabbi Masum Billah and Bradford Campbell. 2019. Unobtrusive Occupancy Detection with FastGRNN on Resource-Constrained BLE Devices. In *Proceedings of DFHS 19: ACM Workshop on Device-Free Human Sensing, New York, NY, USA, November 10, 2019 (DFHS 19),* 5 pages.
https://doi.org/10.1145/3360773.3360874

1 INTRODUCTION

Detecting occupancy in office spaces is a challenging problem, yet a sustainable solution suitable for retrofits would enable a host of applications, from optimized ventilation to more strategic physical space allocation. A recent trend is device-free occupancy detection (DfOD) using RF signals, where wireless devices in the space estimate occupancy and occupants do not need to carry devices or actively participate in the sensing process [3, 10, 13].

DFHS 19, November 10, 2019, New York, NY, USA

ACM ISBN 978-1-4503-7007-3/19/11…$15.00
https://doi.org/10.1145/3360773.3360874

The core idea behind RF-based DfOD is that the presence of humans in an indoor environment affects the received signal strength (RSS), angle of arrival (AoA), and time of flight (ToF) parameters of transmitted RF signals. A typical RF-based DfOD system consists of several radio transmitters and laptop-based receivers, plus an application server which processes the data for human detection [2, 10, 14, 15]. However, the involvement of laptops or Wi-Fi routers in the DfOD system increases the system cost and power requirements as well as deployment complexity. In this work, we focus on making the system scalable, energy-efficient, and practical by lowering the power requirements of the RF devices. This will enable energy-constrained, deployable, and readily-available Bluetooth Low Energy (BLE) devices to perform the sensing task instead of laptops or Wi-Fi routers. However, this presents several challenges as BLE devices often lack computation power, must be frugal with their energy, and lack channel state information (CSI) which state-of-the-art DfOD algorithms typically rely on [16, 19, 21]. Moreover, the majority of existing WiFi-based DfOD systems require multiple measurements and offloading data to the application server, resulting in high energy consumption and communication latency [18, 22, 24, 25].

We present a step towards detecting indoor occupancy using resource-constrained BLE devices without expecting users to have any BLE devices themselves. We deploy four devices in a room, record RSS and a customized ToF measurement for signals between the devices, and use a resource-friendly customized gated recurrent neural network (FastGRNNN [12]) to predict the occupancy of the room. From our initial empirical study, our system can detect the occupancy of a room with 95% accuracy, 96% precision, and 92% recall. In this context, our contribution in this paper is three-fold:

- We develop an occupancy sensing architecture that only requires low-power wireless devices and performs on-board occupancy estimation.
- We estimate the energy required for this approach and demonstrate the feasibility of deployment in indoor settings.
- We identify and discuss potential improvements to this architecture to expand it to additional indoor environments.

2 RELATED WORK

Various approaches for device-free occupancy detection in indoor environments exist with different advantages and limitations. Some systems use multi-sensor data such as CO_2, temperature, and humidity [6, 8] to assess indoor occupancy. However, these approaches are not instantaneous given that the level of CO_2 or temperature alters slowly with respect to human presence. Moreover, these solutions often use various complex machine learning models and need a server to perform the computation [7, 11, 20]. Requiring a server increases the deployment complexity, communication energy consumption, and communication latency.

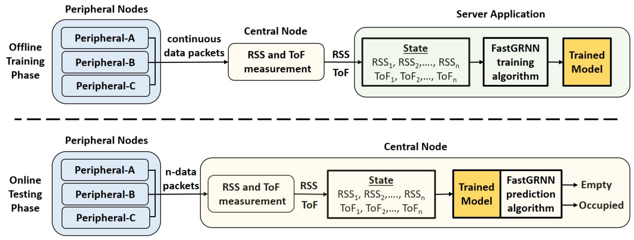


Figure 1: System overview.

Other solutions use non-environmental sensors, such as infrared [5], RFID [23], ultrasonic sound [14], and radio frequency [4, 10, 17]. These solutions, however, can be difficult to deploy. For instance, Karanam et al.'s approach requires three laptops [10], and LiFS requires 11 laptops [17] to perform the human sensing task.

Less sensor-heavy approaches use only commodity Wi-Fi access points to predict the occupancy of a particular room [18, 24, 25]. These solutions may not scale well as deploying one or more access points in every room is costly.

Our proposed approach overcomes the deployment and scalability challenges by using miniature and inexpensive BLE devices. Moreover, this system performs on-device computation and does not require a remote server once deployed. Additionally, since our model uses an RF-based approach it can sample the occupancy of a room on-demand.

3 DESIGN

Our general approach is to deploy N BLE devices (known as peripheral nodes) in the room to be sensed. Each peripheral node interacts with another BLE device, the central node, using standard BLE messages. The central node then uses a machine-learning based model and certain RF properties of those messages to estimate if the room is occupied or not. Our hypothesis is that the presence of occupants will affect the RF signals, and that a data-driven, neural-network based classifier with sufficient training data will effectively detect occupancy even with the limited information provided by commodity BLE devices. As such, our design includes a training phase and a testing phase.

In the offline training phase, each peripheral node repeatedly sends a one byte packet to the central node which measures the corresponding RSS and ToF. Time of flight (ToF) corresponds to the time elapsed between when a message is transmitted and when it is received. While measuring signal strength is straightforward on commodity embedded devices, ToF is more difficult. Our BLE devices do not have synchronized clocks, and as such we adapt a round trip time (RTT) based ToF measurement technique described in [9]. To measure the ToF, the central node sends a message to the peripheral while starting a timer. Upon receiving that signal, the peripheral sends a reply message after a fixed delay. When the central node receives the reply it samples its timer and calculates the propagation time of the signal.

We collect two profiles of training data: a "silent-profile" when the room is empty, and a "noise-profile" when there is at least one person in the room. The central node forwards the raw data from these profiles to a remote server to train a kilobyte-sized recurrent neural network (RNN).

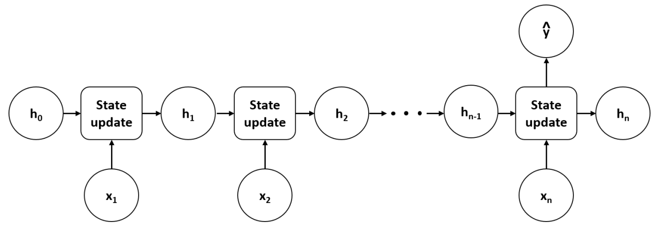


Figure 2: Learning process block diagram.

In the testing phase, we deploy and execute the trained model on the central node. Peripheral nodes once again send packets to the central node and the central node predicts the occupancy of the room using the measured RSS and ToF. The overall system flow is depicted in Figure 1.

3.1 Machine Learning Algorithm

Our architecture uses a fast and tiny gated recurrent neural network (FastGRNN) [12]. While many machine learning techniques (RNN, LSTM, GRU) exist that might be effective for our data, they typically generate a large model which cannot fit into the memory-constrained central node. Compared to a conventional RNN algorithm, the FastGRNN model has lower prediction costs (18x faster) and a much smaller memory footprint (35x smaller) [12].

3.1.1 Notation. Table 1 shows the notation used to describe the training and prediction process. The input feature vector collected at i-th time step is defined by $\mathbf{x}_i \in \mathbb{R}^d$. Here, d is the dimension of the feature vector, which in our case is six, as we use two features (RSS and ToF) from each of the three peripherals. As such, for peripherals A, B and C at i-th step we get $\mathbf{x}_i = \{RSS_A, ToF_A, RSS_B, ToF_B, RSS_C, ToF_c\}$ as the input feature vector.

RNNs maintain a hidden state vector $\mathbf{h}_i \in \mathbb{R}^{\hat{d}}$ for each time-step i to allow information to flow from the previous step to the next step. In each iteration of the training algorithm, our objective is to optimize matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$; vector $\mathbf{b}$; scalar values α and β. Hyperparameters r_1, r_2 and r_3 are used to control the size of the model using low rank decomposition of matrices $\mathbf{U}$ and $\mathbf{V}$ and $\mathbf{W}$.

Table 1: Notation

Symbol	Description
$\boldsymbol{x_i}$	Feature vector at i-th time-step. Contains RSS and ToF of three peripherals
$\boldsymbol{h_i}$	Hidden state vector at i-th step
$\mathbf{U}, \mathbf{V}, \mathbf{W}$	RNN learning matrices, tunes the model
$\boldsymbol{b}$	Bias vector
α, β	Trainable scalar weights
n	Number of time-steps. Consecutive signals fed into the learning model at one iteration
d	Dimension of the feature vector $\boldsymbol{x}$
$\hat{d}$	Dimension of the hidden state vector $\mathbf{h}$
r_1, r_2, r_3	Hyperparameters, together control the dimension of $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$

3.1.2 Learning Process. Figure 2 illustrates a block diagram of our learning process. Let, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ be the set of input features that we are feeding to the FastGRNN model at once, where $\mathbf{x}_i$ feature vector is collected immediately before the $\mathbf{x}_{i-1}$ feature

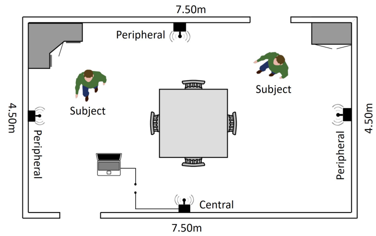


Figure 3: A sample testbed where we collect data using four nRF52840 BLE devices

vector. n represents the number of time-steps i.e. number of times information to be passed to the next step of the network. Increasing n should improve accuracy, but at the expense of needing to capture and process more RF signals, which increases energy consumption and measurement latency. From our empirical study, taking n = 3 provides us satisfactory result (i.e. 94% accuracy), but we also explore tuning this parameter.

The state update block of Figure 2 provides us the updated state vector $\mathbf{h}$ for the next phase. Inside of this block following operation is being performed to compute the next hidden state.

$$\mathbf{g}_i = \tanh(\mathbf{W}\mathbf{x}_i + \mathbf{U}\mathbf{h}_{i-1} + \mathbf{b}) \tag{1}$$

$$\mathbf{h}_i = (\alpha(1 - \mathbf{g}_i + \beta) \odot \mathbf{g}_i + \mathbf{g}_i \odot \mathbf{h}_{i-1} \tag{2}$$

where, $0 \leq \alpha, \beta \leq 1$ are trainable parameters and $\odot$ stands the Hadamard product.

Once we have the final hidden state vector $\mathbf{h}_n$, we can predict the output value using following equation-

$$\hat{y} = softmax(\mathbf{V}\mathbf{h}_n) \tag{3}$$

In regular RNN model, parameter matrices $\mathbf{U} \in \mathbb{R}^{\hat{d}\times\hat{d}}$, $\mathbf{V} \in \mathbb{R}^{\hat{d}\times\hat{d}}$ and $\mathbf{W} \in \mathbb{R}^{\hat{d}\times d}$ are quite large, which makes it difficult to fit into BLE devices. However, in FastGRNN, $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ are compressed using low-rank matrix decomposition as follows-

$$\mathbf{U} = \mathbf{U}_1(\mathbf{U}_2)^T; \mathbf{V} = \mathbf{V}_1(\mathbf{V}_2)^T; \mathbf{W} = \mathbf{W}_1(\mathbf{W}_2)^T; \tag{4}$$

where, $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{\hat{d}\times r_2}$; $\mathbf{V}_1, \mathbf{V}_2 \in \mathbb{R}^{\hat{d}\times r_3}$; $\mathbf{W}_1 \in \mathbb{R}^{\hat{d}\times r_1}$ and $\mathbf{W}_2 \in \mathbb{R}^{d\times r_1}$. Controlling the ranks r_1, r_2 and r_3 we trade-off between the model size and model performance.

Once we determine the output using equation 3, we calculate the loss L on this prediction (logistic loss), and using the mini-batch stochastic gradient descent we jointly update our learning parameters $\boldsymbol{\theta} = \{\mathbf{U}_1, \mathbf{U}_2, \mathbf{V}_1, \mathbf{V}_2, \mathbf{W}_1, \mathbf{W}_2, \mathbf{b}, \alpha, \beta\}$.

The result of feeding the entire training dataset into the model in an iterative fashion is an optimized trained parameter set $\boldsymbol{\theta}$. We then test this with our offline testing dataset and verify that the model performs well. Finally, we deploy these optimized parameters into the central node for the online testing phase. We take three consecutive signals from each of the peripheral nodes and feed them into the model as feature vectors $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. Using equations (1), (2), and (3) the central node predicts the occupancy.

3.2 Testbed

Our system uses four BLE nodes, three as peripheral nodes and one as the central node. Figure 3 shows our testbed setup. Once deployed, the peripheral nodes start advertising and the central node scans and establishes connections with each peripheral node. After collecting enough measurements, the central node runs the model and estimates occupancy. For our experiments, the prediction is sent over a serial connection to a laptop for analysis.

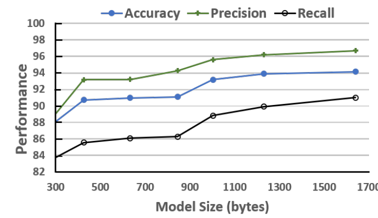


Figure 4: Performance-memory trade-off. Increasing the model size improves performance, but a model larger than few kilobytes will not fit on the nRF52840.

4 EXPERIMENTAL EVALUATION

Our experiments evaluate the system sensitivity to different parameter settings, trade-off between performance and device memory, and the estimated lifetime of the system in practical deployment. Our experiments answer the following queries:

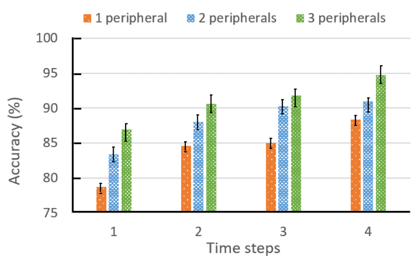
- How accurate is our system in a practical deployment? Results suggest that using this model we can detect occupancy with up-to 95% accuracy.
- How sensitive is our model with different parameter settings, such as the number of peripherals, number of messages, size of the trained model?
- How long will the system run once deployed? We show that, using a 235 mAh, 3V lithium battery this system can perform up-to 6 years.

4.1 Experimental Setup

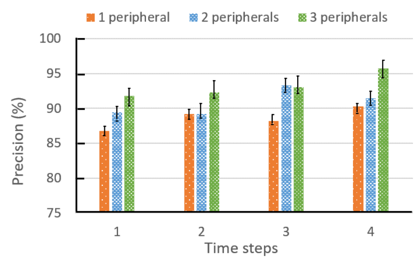
We evaluate the system using four Nordic nRF52840 BLE development kits which is built around a 32-bit, 64MHz Arm Cortex-M4F CPU with 1MB flash memory and 256 kB RAM. Our program is tested using Nordic's Softdevice 140 v5.0.0-2alpha BLE software core.

We perform our experiments in two different rooms with dimensions 4.5 m × 7.5 m and 4 m × 6 m, respectively. These rooms include their usual furniture. We attach four BLE devices three feet above the ground as illustrated in Figure 3. In the training phase, our system collects data for approximately 35 minutes when the room is empty. After that, a person walks in the room, pauses in different positions and the central node repeats the data collection process for about 30 minutes. Finally, two people walk in the room and the process continues for another 30 minutes. In this way, we collect approximately 20,000 data points. We subdivide 80% data for training the machine learning model and 20% data for offline verification of the model.

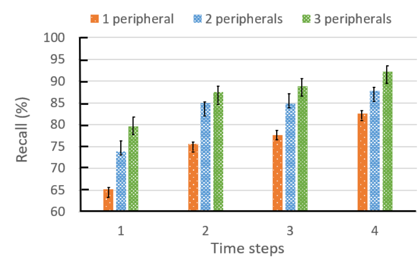
Once the model has been trained, in online testing phase we deploy the model on the central node. The central node collect data for one hour, with the room being 40 minutes occupied and 20 minutes unoccupied. The occupants perform usual activities such as sitting, eating, or walking around. Using the model the central node predicts whether the room is empty or not for each captured data point.



(a) Accuracy



(b) Precision



(c) Recall

Figure 5: System performance.

4.2 System performance

We use three metrics to evaluate the performance of occupancy detection.

- **Accuracy:** Overall, how often does the system make correct prediction?
- **Precision:** When the system predicts occupied, how often it is correct?
- **Recall:** When the room is actually occupied, how often does the system correctly identify that?

Figure 5 shows our accuracy, precision, and recall results. From our observation, we can improve performance by increasing the number of peripherals and the number of time-steps of the model. We reach maximum performance with 95% accuracy, 96% precision, and 92% recall using three peripherals and four consecutive time-steps.

4.3 Accuracy-Memory Trade-Off

Figure 4 illustrates the performance of the system as we vary the size of the deployed model. Referring to Equation 4, we can change the size of the model controlling ranks r_1, r_2 and r_3. Increasing the size of the model makes the learning matrices less sparse and less quantized, which results in improved accuracy. Notice that the system achieves 95% accuracy using only a 1.6 kB model when we take hyperparameters $r_1 = 6$, $r_2 = 4$ and $r_3 = 4$.

4.4 System Lifetime

One of the main challenges of using WiFi-based DfOD approaches is their high energy requirements. To make RF-based occupancy detection sensors deployable they must last multiple years with reasonably sized batteries.

Figure 6 illustrates the estimated current draw of the central node during one cycle of the testing mode. We use the "Nordic power profile" simulation tool [1] to collect this data for our application. The "Data collection" portion of the figure shows the average current required for one TX byte, radio switch, and one RX byte cycle repeated nine times (three packets × three peripherals). The "Post processing" segment indicates the average current for measuring RSS and ToF, and predicting the room occupancy. The "Keep alive" segment results from the nRF52840 requiring the central node to communicate with each peripheral at least every 4,000 ms to keep the connection alive. Our approach can perform a measurement as needed. To estimate average current, we assume a measurement is taken every 30 seconds and the node is in sleep mode (at 2 μA) otherwise. From our analysis, the central node will draw 4.42 μA on average in one cycle (30010 ms) of occupancy detection. As such, if we are using a 235 mAh, 3V Lithium battery in the BLE board our system will have:

$$\text{battery life} = \frac{235 \text{ mAh} \times 3 \text{ V}}{4.42\mu\text{A} \times 3 \text{ V}} \approx 159,864 \text{ hrs} \approx 6 \text{ years}$$

The estimated average current of the simulation tool for nRF52840 board is typically within 5% of the actual value. As such, we can infer that the system will sustain from 5.7 years to 6.3 years.

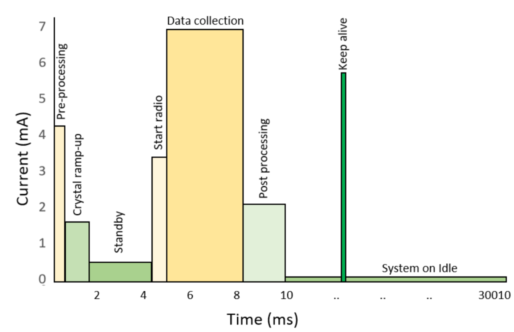


Figure 6: Current profile over one cycle. On average 4.42 μA current is required to complete a cycle.

5 DISCUSSION AND CONCLUSION

Our implementation and experiments show that it is feasible to predict the occupancy in an indoor environment using miniature, easily deployable and energy-frugal BLE devices. However, we have trained and tested our system using limited number of rooms. We intend to expand this system in several directions including training and testing the system in more diverse environment, involving more subjects, capturing real-world power traces, and eventually predicting the number of people.

Traditional fingerprinting based approaches in DfOD have a drawback that they are environment-specific and do not perform well in environments where they have not been trained. We intend to explore resolving this by leveraging reinforcement learning (RL) and feedback from other IoT devices commonly used in smart buildings that accelerate with human activity. This should further improve the deployability of DfOD to make it truly viable in real buildings.

REFERENCES

[1] 2019. Online power profiler. https://devzone.nordicsemi.com/nordic/power. (2019). Accessed: 08-06-2019.

[2] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. 2015. Capturing the human figure through a wall. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 219.

[3] Fadel Adib, Zachary Kabelac, and Dina Katabi. 2015. Multi-person localization via {RF} body reflections. In *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*. 279–292.

[4] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C Miller. 2014. 3D tracking via body radio reflections. In *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*. 317–329.

[5] Yuvraj Agarwal, Bharathan Balaji, Seemanta Dutta, Rajesh K Gupta, and Thomas Weng. 2011. Duty-cycling buildings aggressively: The next frontier in HVAC control. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE, 246–257.

[6] Colin Brennan, Graham W Taylor, and Petros Spachos. 2018. Designing learned CO2-based occupancy estimation in smart buildings. *IET Wireless Sensor Systems* 8, 6 (2018), 249–255.

[7] H Elkhoukhi, Y NaitMalek, A Berouine, Mohamed Bakhouya, D Elouadghiri, and Mohammed Essaaidi. 2018. Towards a real-time occupancy detection approach for smart buildings. *Procedia computer science* 134 (2018), 114–120.

[8] Alessandro Franco, Francesco Leccese, and Lorenzo Marchi. 2019. Occupancy modelling of buildings based on CO2 concentration measurements: an experimental analysis. In *Journal of Physics: Conference Series*, Vol. 1224. IOP Publishing, 012016.

[9] Davide Giovanelli and Elisabetta Farella. 2018. Rssi or time-of-flight for bluetooth low energy based localization? an experimental evaluation. In *2018 11th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 1–8.

[10] Chitra R Karanam, Belal Korany, and Yasamin Mostofi. 2019. Tracking from One Side-Multi-Person Passive Tracking with WiFi Magnitude Measurements. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 181–192.

[11] Wilhelm Kleiminger, Christian Beckel, Thorsten Staake, and Silvia Santini. 2013. Occupancy detection from electricity consumption data. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. ACM, 1–8.

[12] Aditya Kusupati, Manish Singh, Kush Bhatia, Ashish Kumar, Prateek Jain, and Manik Varma. 2018. Fastgrnn: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network. In *Advances in Neural Information Processing Systems*. 9017–9028.

[13] Xiang Li, Shengjie Li, Daqing Zhang, Jie Xiong, Yasha Wang, and Hong Mei. 2016. Dynamic-music: accurate device-free indoor localization. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 196–207.

[14] Wenguang Mao, Mei Wang, and Lili Qiu. 2018. Aim: acoustic imaging on a mobile. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 468–481.

[15] Ahmed Saeed, Ahmed E Kosba, and Moustafa Youssef. 2013. Ichnaea: A low-overhead robust WLAN device-free passive localization system. *IEEE Journal of selected topics in signal processing* 8, 1 (2013), 5–15.

[16] Shuyu Shi, Stephan Sigg, Lin Chen, and Yusheng Ji. 2018. Accurate location tracking from CSI-based passive device-free probabilistic fingerprinting. *IEEE Transactions on Vehicular Technology* 67, 6 (2018), 5217–5230.

[17] Ju Wang, Hongbo Jiang, Jie Xiong, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Binbin Xie. 2016. LiFS: low human-effort, device-free localization with fine-grained subcarrier information. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 243–256.

[18] Junqi Wang, Norman Chung Fai Tse, and John Yau Chung Chan. 2019. Wi-Fi based occupancy detection in a complex indoor space under discontinuous wireless communication: A robust filtering based on event-triggered updating. *Building and Environment* 151 (2019), 228–239.

[19] Kaishun Wu, Jiang Xiao, Youwen Yi, Dihu Chen, Xiaonan Luo, and Lionel M Ni. 2012. CSI-based indoor localization. *IEEE Transactions on Parallel and Distributed Systems* 24, 7 (2012), 1300–1309.

[20] Zheng Yang, Nan Li, Burcin Becerik-Gerber, and Michael Orosz. 2014. A systematic approach to occupancy modeling in ambient sensor-rich buildings. *Simulation* 90, 8 (2014), 960–977.

[21] Zheng Yang, Zimu Zhou, and Yunhao Liu. 2013. From RSSI to CSI: Indoor localization via channel response. *ACM Computing Surveys (CSUR)* 46, 2 (2013), 25.

[22] Moustafa Youssef, Matthew Mah, and Ashok Agrawala. 2007. Challenges: device-free passive localization for wireless environments. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM, 222–229.

[23] Han Zou, Lihua Xie, Qing-Shan Jia, and Hengtao Wang. 2014. Platform and algorithm development for a rfid-based indoor positioning system. *Unmanned Systems* 2, 03 (2014), 279–291.

[24] Han Zou, Yuxun Zhou, Jianfei Yang, Weixi Gu, Lihua Xie, and Costas Spanos. 2017. Freedetector: Device-free occupancy detection with commodity wifi. In *2017 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*. IEEE, 1–5.

[25] Han Zou, Yuxun Zhou, Jianfei Yang, and Costas J Spanos. 2018. Device-free occupancy detection and crowd counting in smart buildings with WiFi-enabled IoT. *Energy and Buildings* 174 (2018), 309–322.