Pi-NIC: Indoor Sensing Using Synchronized Off-The-Shelf Wireless Network Interface Cards and Raspberry Pis

Fangzhan Shi*, Wenda Li*, Amin Amiri†, Shelly Vishwakarma*, Chong Tang*, Paul Brennan†, Kevin Chetty*
*Department of Security and Crime Science, University College London, UK
†Department of Electronic and Electrical Engineering, University College London, UK
{fangzhan.shi.17, wenda.li, amin.amiri, s.vishwakarma, chong.tang.18, p.brennan, k.chetty}@ucl.ac.uk

***Abstract*—This paper presents an indoor joint communication and sensing system that consists of synchronized off-the-shelf wireless network interface cards (NIC) and Raspberry Pis. There exists a significant body of research that uses the channel state information (CSI) reported by wireless network interface cards for sensing, but only the amplitude and phase difference of the CSI between receiver antennas are processed. The raw phase of the CSI is contaminated by the carrier frequency offset, packet detection delay and other hardware imperfections, so it is too noisy to use. Our work introduces the raw phase of CSI into sensing by synchronizing the transmitter and receiver clocks to remove carrier frequency offset and using a new method to remove packet detection delay. We validate our design in a real-world scenario to detect breathing and walking and demonstrate that the raw phase of the CSI offers an evident improvement in Wi-Fi CSI-based sensing. Additionally, we are the first to use the Raspberry Pi and ATH9k wireless network interface card together for CSI data collection, which is cheap, portable and versatile.**

***Index Terms*—Joint Communications and Sensing (JCAS), Channel State Information (CSI), Synchronization, Raspberry Pi**

I. Introduction

The concept of joint communication and sensing (JCAS) has been growing in recent years and indoor sensing using Wi-Fi signal has attracted attention from researchers as it shows applications in health care [1]–[4], activity recognition [5]–[7], localization [8]–[10] and security [11], [12]. The major advantage of this type of sensing is that Wi-Fi signal is ubiquitous, pervasive and sensing using it is privacy-preserving. Moreover, it does not require light can even provide through-the-wall sensing.

There are two major approaches to building the sensing system. One is to use software-defined radio (SDR) like the well known universal software radio peripheral (USRP) [4], [5], [8], [11] and the other is to use commercial-level off-the-shelf equipment [1]–[3], [6], [7], [9], [10]. The SDR approach is based on the principle of bistatic radar that the Wi-Fi signal transmitter is treated as a source of RF illumination and the SDR is treated as a spatially separated receiver. The cross ambiguity function [12] is widely used here to find the Doppler frequency shift and range. Using the off-the-shelf equipment, one or multiple off-the-shelf wireless network interface cards (Wi-Fi cards) are used with regular computers to collect the data. The data is the CSI that describes the channel response and it can be used for sensing. However, typically, the NIC would not report the data to the user. Special models of NIC and corresponding software are required to output the CSI from NIC. For example, [13] can only be used on Intel NIC 5300, the Atheros CSI Tool [14] is designed for the ATH9k family Wi-Fi chips, Nexmon [15] can run on several Broadcom Wi-Fi chips.

Both of the approaches have advantages and disadvantages. For the SDR, bistatic radar signal processing has been well researched and the techniques are mature. The achievable resolution is satisfactory and machine learning methods can be used to further enhance its capabilities. However, it is costly for commercial applications or distributed sensing systems. Additional computers are required to control the devices and process data which adds extra cost. For the commercial devices, due to the hardware limitation, the accuracy of the CSI data is low, especially for the phase of CSI values. Workarounds and complicated algorithms are developed to ease this problem and make the CSI useful for sensing but a number of limitations and restrictions are applied, such as limiting the target sensing area. In terms of cost, NIC's are significantly more affordable. Note that both approaches require computers which limits the portability of the sensing system.

Compared to previous work [1]–[3], [6], [7], [9], [10], the following contributions are made by this paper: We improve the sensing capability of wireless NICs by synchronization and a packet detection delay removal method. It includes modification to the NIC hardware and developing corresponding software. Additionally, we test several signal processing methods that cannot be used in previous research due to hardware limitations and the result demonstrates breathing detection and walking detection with increased sensitivity and specificity.

The following paper is organized as follows: we describe the principle of CSI sensing and its measurement limitations in section II and introduce our approaches to solve the solve carrier frequency offset and packet detection delay in section III. We then show our testbed and experimental results in

Fangzhan Shi is funded by the China Scholarship Council (CSC) from the Ministry of Education of P.R. China

section IV. Finally, we conclude in Section V and discuss future avenues of work.

II. CSI-Based Wi-Fi Sensing

A. Idealized CSI Sensing Model

Channel state estimation is one critical step to demodulate OFDM signals as it can remove the adverse effect in real-world channels such as frequency selective fading. In 802.11n [16] OFDM Wi-Fi packets, CSI is calculated from fixed preambles defined in the protocol. The CSI is the output of channel state estimation that describes the frequency response of the channel. To present the frequency response, each value in the CSI is complex whose magnitude and phase indicate the amplitude response and phase response respectively. There are $N_{tx} \times N_{rx} \times N_c$ complex values in one CSI record where N_{tx} is the number of antennas on the transmitter, N_{rx} is the number of antennas on the receiver and N_c is the number of subcarriers.

Multipath exists in real-world channels due to surrounding objects and the presence of people. Suppose there are one transmitter antenna and one receiver antenna, we can write the channel state information as follows [14]:

$$H(f_{bb}, t) = \sum_{k=1}^{N} a_k(f_{bb}, t) e^{-j2\pi(f_c + f_{bb})\tau_k(t)} \tag{1}$$

where f_c is the center carrier frequency, f_{bb} is the baseband frequency of the subcarrier, t is the time, N is the number of paths, $a_k(f, t)$ is the complex number representing the attenuation and initial phase offset of the channel, $\tau_k(t)$ is propagation delay and $n(t)$ is the additive white Gaussian noise.

If we simplify the channel so that there are only two signal paths, the line-of-sight (los) signal and the reflection signal from the human body, the CSI would be:

$$\begin{aligned} H'(f_{bb}, t) = \quad & a_{los}(f_{bb}, t) e^{-j2\pi(f_c + f_{bb})\tau_{los}(t)} \\ & + a_{body}(f_{bb}, t) e^{-j2\pi(f_c + f_{bb})\tau_{body}(t)} \end{aligned} \tag{2}$$

By processing the $H'(f_{bb}, t)$, sensing can be achieved and applications can be developed [17].

B. Phase Issue in CSI

1) Carrier Frequency Offset (CFO): The NIC uses a phase lock loop (PLL) to synthesize the Wi-Fi carrier signal [18]. The carrier signal frequency is generated as follows:

$$f_c = \frac{M}{N} f_{osc} \tag{3}$$

where f_{osc} is the local crystal oscillator frequency and M, N are programmable integer values. For example, if the f_{osc} is 40MHz, to generate a carrier frequency of 5825MHz, M and N could be set to 1165 and 8 respectively.

As the crystals on the transmitter and the receiver are not identical, they would oscillate at different frequencies. Using the previous example of PLL and assuming the crystal frequency difference is 25 parts per million (ppm), the carrier frequency offset Δf_c would be 145.625kHz. The NIC could deal with this error for communications using a frequency correction algorithm based on the preambles in Wi-Fi frame but the error is still too large for the purpose of sensing and it will continually introduce a fluctuating CSI phase [19]. Based on this model, the CSI would be

$$H''(f_{bb}, t) = e^{-j2\pi \int \Delta f_c(t) dt} H'(f_{bb}, t) \tag{4}$$

$\Delta f_c(t)$ is the instantaneous carrier frequency offset and it could change over time due to hardware imperfections.

2) Packet Detection Delay (PDD): The NIC would calculate CSI after a packet is detected [19]. We define the packet detection delay τ_{pdd} as the time when the receiver detects the packet subtracted by the time when the transmitter sends the signal. As the baseband is based on digital sampling, τ_{pdd} would be dependent on the baseband sampling clock on the transmitter and receiver.

$$H'''(f_{bb}, t) = e^{-j2\pi\tau_{pdd} f_{bb}} H''(f_{bb}, t) \tag{5}$$

The raw phase of CSI is not very useful for sensing as both of CFO and PDD will introduce uncertainty to it, but we can cancel the phase issue by using the power of CSI $H(f, t)H(f, t)^*$ (* represents conjugate) [2], [3] or use the conjugate multiplication of CSI between two antennas on one receiver $H_{ant1}(f, t)H_{ant2}(f, t)^*$ [7], [9], [20] as they share the same CFO and PDD. By processing the Δf_c-free term over time, we can achieve activity recognition and localization.

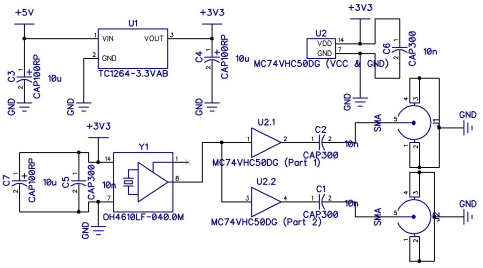
III. Solution To Phase Issue

A. CFO Removal

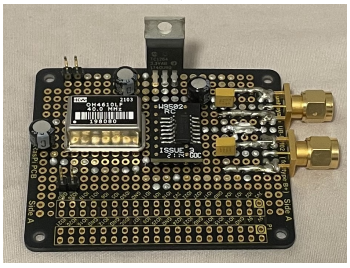
Simon and Aydin [21] synchronized two wireless routers with ATH9k Wi-Fi chips using one shared clock to increase the accuracy of angle of arrival (AoA) estimation. In their work, synchronization permits them to increase the number of antennas in the phased array. Moreover, the signal processing is still based on the phase difference between antennas as all the wireless routers are configured as receivers. It inspired us that by synchronizing transmitter and receiver, we can use the raw phase of CSI directly and it could provide a richer measurement of the actual channel.

We designed and made a circuit board for the oscillator as Fig.1. It is made up of an active crystal (Y1) and digital buffers (U2). The active crystal can generate 40MHz square wave continuously and it is the actual clock source in the synchronization. Digital buffers are used to split the crystal output into two outputs and improve the output current. Two capacitors (C1, C2) are used as DC-block before the SMA connectors as we found the chip would provide a DC bias voltage by itself. Voltage regulator (U1) is used to provide a stable 3.3v power source and extra capacitors (C3, C4, C5, C6, C7) can filter undesired noise from the power supply.

We choose to use ATH9k wireless network interface card and the specific model is AR9388. The NIC does not support an external clock input, so we need to modify the hardware to inject the external clock into the chip. The crystal is removed and one coaxial cable with a stripped end is soldered on the

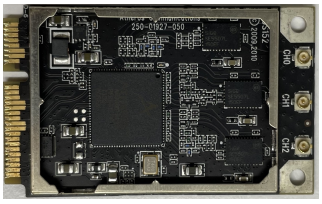


(a)

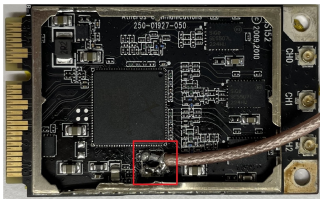


(b)

Fig. 1: Oscillator board: (a) schematic diagram (b) actual board



(a) Original



(b)

Fig. 2: NIC card: (a) original (b) modified (wired connection in red box)

corresponding pads as the external clock input port. Fig.2 shows the modification. Using synchronization, Δf_c can be set to 0 in theory.

B. PDD Removal

A new method to remove PDD is developed. Note that the baseband subcarrier frequencies are symmetric, which means for a subcarrier f_{bb}, we can always find another subcarrier $-f_{bb}$ in a 802.11n OFDM packet and all subcarriers in the packet shares the same PDD, the product of the CSI for subcarrier f_{bb} and subcarrier $-f_{bb}$ is PDD-free as

$$\begin{aligned} H'''(f_{bb},t)H'''(-f_{bb},t) = \quad & e^{-j2\pi\tau_{pdd}f_{bb}}H''(f_{bb},t) \\ & \times e^{-j2\pi\tau_{pdd}-f_{bb}}H''(-f_{bb},t) \\ = \quad & H''(f_{bb},t)H''(-f_{bb},t) \end{aligned} \tag{6}$$

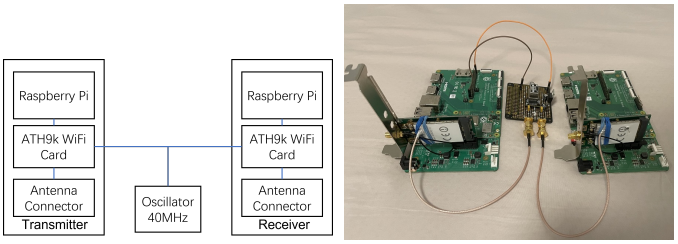
Since Δf_c has been set to 0 and $f_c >> f_{bb}$, the product can be approximated as Equation.7. As the line-of-sight channel generally would not change over time but the body-reflection channel is time-varying due to any motion, $a_{los}(f_{bb},t)e^{-j2\pi f_c\tau_{los}(t)}a_{los}(-f_{bb},t)e^{-j2\pi f_c\tau_{los}(t)}$ can be seen as the only DC component in the product and it is represented as C. Assuming the signal from line-of-sight channel is much stronger, $a_{body}(f_{bb},t)e^{-j2\pi f_c\tau_{body}(t)}a_{body}(-f_{bb},t)e^{-j2\pi f_c\tau_{body}(t)}$ would be the smallest term and it can be ignored. If the following post-processing methods are window-based (such as short-time Fourier transformation) and the window size is relatively small, $a_{body}(f_{bb},t)$ and $a_{body}(-f_{bb},t)$ can be treated as constants in the window. The overall approximation would be $Ke^{-j2\pi f_c\tau_{body}(t)}+C$ and K is a constant factor. From now on, CFO and PDD are removed.

$$\begin{aligned} H'''(f_{bb},t)H'''(-f_{bb},t) = \quad & H'(f_{bb},t)H'(-f_{bb},t) \\ = \quad & a_{los}(f_{bb},t)e^{-j2\pi(f_c+f_{bb})\tau_{los}(t)}a_{los}(-f_{bb},t)e^{-j2\pi(f_c-f_{bb})\tau_{los}(t)} \\ & +a_{body}(f_{bb},t)e^{-j2\pi(f_c+f_{bb})\tau_{body}(t)}a_{body}(-f_{bb},t)e^{-j2\pi(f_c-f_{bb})\tau_{body}(t)} \\ & +a_{los}(-f_{bb},t)e^{-j2\pi(f_c-f_{bb})\tau_{los}(t)}a_{body}(f_{bb},t)e^{-j2\pi(f_c+f_{bb})\tau_{body}(t)} \\ & +a_{los}(f_{bb},t)e^{-j2\pi(f_c+f_{bb})\tau_{los}(t)}a_{body}(-f_{bb},t)e^{-j2\pi(f_c-f_{bb})\tau_{body}(t)} \\ \approx \quad & a_{los}(f_{bb},t)e^{-j2\pi f_c\tau_{los}(t)}a_{los}(-f_{bb},t)e^{-j2\pi f_c\tau_{los}(t)} \\ & +a_{body}(f_{bb},t)e^{-j2\pi f_c\tau_{body}(t)}a_{body}(-f_{bb},t)e^{-j2\pi f_c\tau_{body}(t)} \\ & +a_{los}(-f_{bb},t)e^{-j2\pi f_c\tau_{los}(t)}a_{body}(f_{bb},t)e^{-j2\pi f_c\tau_{body}(t)} \\ & +a_{los}(f_{bb},t)e^{-j2\pi f_c\tau_{los}(t)}a_{body}(-f_{bb},t)e^{-j2\pi f_c\tau_{body}(t)} \\ \approx \quad & Ke^{-j2\pi f_c\tau_{body}(t)}+C \end{aligned} \tag{7}$$

IV. Testbed and Experimental Result

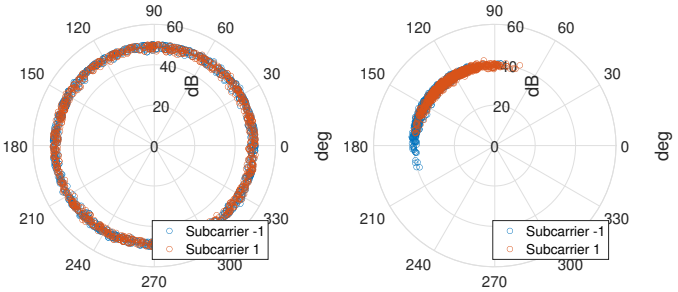
A. Testbed

Raspberry Pi Compute Module 4 is utilised with the corresponding IO Board Raspberry Pi to support a PCIe x1 interface. The ATH9k official driver is available in the Raspberry Pi OS Linux Kernel source, but it does not support the CSI-related functions because this function is not essential for general users. We modified the official driver by adding those functions from the Atheros CSI Tool [14] and rewriting the obsolete APIs to make it work on the Raspberry Pi. Additionally, power save mode is disabled to keep the PLL



(a) (b)

Fig. 3: Overview of synchronized devices: (a) block Diagram (b) actual devices



(a) (b)

Fig. 4: CSI in polar coordinates: (a) with asynchronized clock source and (b) with synchronized clock source. Each circle represents one CSI record.

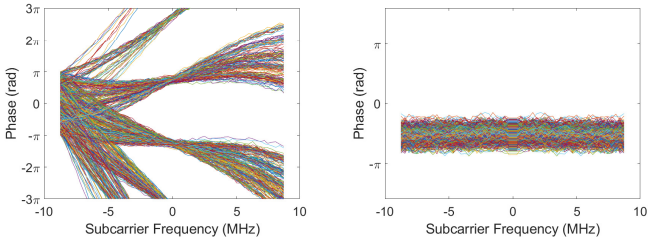
working all the time. Fig.3 illustrates the whole system, including the Raspberry Pis, wireless interface cards and the shared clock.

The NIC is set to communicate at channel 165 (5.825GHz) using 20MHz bandwidth during all experiments. The transmitter is configured to send 1000 802.11n packets per second to the receiver at a power of 15dBm. One transmitter antenna and one receiver antenna are enabled. In this scenario, each CSI record is $1 \times 1 \times 56$ complex numbers.

B. Effectiveness of Proposed Concepts

We connect the transmitter and the receiver via a coaxial cable and a -20dB attenuator to validate our design and compare it to the asynchronized system. For each experiment, 1000 consecutive CSI records are collected and analyzed.

1) CFO Removal: According to Equation.5, the phase of CSI would be affected by packet detection delay. As a result, to validate the CFO removal only, we should use the subcarrier with minimum absolute baseband frequency. Since subcarrier 0 is not used according to the 802.11n protocol, the phase of subcarrier -1 and subcarrier 1 are analyzed. In Fig.4, we can see the phase of CSI covers the entire phase space for the asynchronized devices but there is only one phase cluster for the synchronized devices. This result provides validation of our synchronization method. The phase plot is not ideal as the cluster for each subcarrier is around 60 degs wide, which means the two devices are not perfectly synchronized due to clock noise and jitters.



(a) (b)

Fig. 5: CSI Phase (Synchronized NICs): (a) without PDD removal (b) with PDD removal. Different CSI records are represented by different curves.

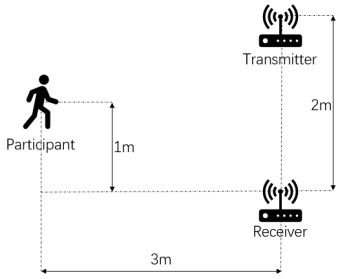


Fig. 6: Test Environment

2) PDD Removal: Theoretically, the plot of the unwrapped phase should be linear with respect to the subcarrier frequency and the slope is a result of the delay. From Fig.5, curves' approximated linear slopes show many clusters due to the existence of PDD, but after applying PDD removal, there is only one horizontal cluster remaining. Additionally, the non-linearity of the phase curve due to hardware imperfection is also removed using the PDD removal method. It is because the major non-linear distortion is symmetric with respect to subcarrier 0 and shows similar property as PDD.

C. Experiments Using Real-world Indoor Channel

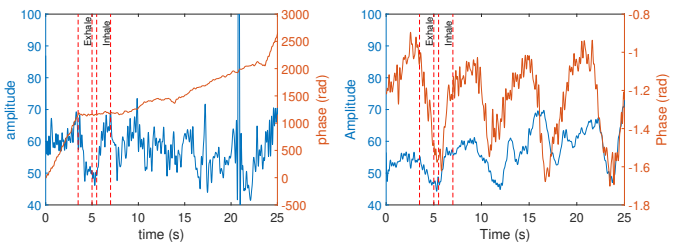
Initial testing was carried out in an indoor environment and geometry is shown in Fig.6.

1) Breathing Detection: Expansion and contraction of the chest wall when breathing regularly is expected to give rise to a clear periodic pattern in both the CSI amplitude and phase [2], [3]. There are 56 subcarriers and we merge them into one for simplicity as follows:

$$amplitude(t) = LPF(\frac{1}{N}\sum_{i=1}^{N} abs(H_m(f_{bbi}, t))) \quad (8)$$

$$phase(t) = LPF(unwrap(angle(\frac{1}{N}\sum_{i=1}^{N} \frac{H_m(f_{bbi}, t)}{abs(H_m(f_{bbi}, t))})) \quad (9)$$

where H_m is the measured CSI, N is the number of subcarriers, f_{bbi} is the frequency of i^{th} subcarrier, LPF is a low-pass filter whose cutoff frequency is 5 Hz, $angle$ is the function to calculate the angle of a complex number and



(a) (b)

Fig. 7: CSI for breathing detection: (a) with asynchronized clock (b) with synchronized clock

$unwrap$ is the function to unwrap angles to avoid $-\pi$ to π jump. When using asynchronized NICs, according to Fig.7a, the CSI amplitude can show the pattern but the phase keeps increasing showing no breathing pattern. For the synchronized NICs, phase changes matches the amplitude changes exactly in Fig.7b and both of them can be used for breathing detection. Note that the changing tendency of amplitude and phase is not necessarily to be the same because it will vary from the position of the target.

2) Walking Detection: Walking forward and backward is a relatively fast and large body movement. In this case, the processing is similar to conventional Doppler frequency shift analysis. As there are 56 subcarriers available, we process each subcarrier respectively then merge the spectrograms to one as follows:

$$spectro = \frac{1}{N}\sum_{i=1}^{N} 20\log(abs(STFT(HPF(H'_m(f_{bb_i}, t))))) \tag{10}$$

where $H'_m(f_{bb_i}, t)$ is the pre-processed CSI, HPF is a high pass filter whose cutoff frequency is 3Hz, $STFT$ is short-time Fourier transformation with a 0.512s window and 0.384 overlap.

Since CSI amplitude-based methods has been widely used in previous research [17] [22], we apply CFO removal and PDD removal incrementally then extract CSI amplitude as the pre-processing. In Fig.8, as the CFO removal and PDD removal methods only work for the phase, it would not change the spectrogam of the CSI amplitude and all of the three spectrograms show very similar patterns. Note that it cannot distinguish if the target is moving forward or backward directly from the frequency as both of positive and negative frequencies exist in the spectrograms.

As CFO removal and PDD removal are available, we can perform the previous pre-processing without extracting the CSI amplitude and use the raw values to plot the spectrograms. Fig.9 shows that using CFO removal alone can improve the signal-to-noise ratio significantly but there still exists an evident noise, especially, the spectrogram mirrors. The following PDD removal can refine it and provide a better result. It shows the Doppler frequency shift on the spectrogram and when the target is approaching, there would be a positive frequency shift, vice versa.

The maximum speed of the target is measured as 1.6m/s and according to the geometry, the bistatic speed is 3.0m/s, leading to a theoretical Doppler frequency shift of 58 Hz. From the spectrogram, the maximum Doppler frequency shift is about 60Hz and it matches the theoretical calculation.

V. Conclusion and Future Work

The work presented in this paper highlights how future communications infrastructures may employ CSI for enhanced joint communications and sensing. By synchronizing two off-the-shelf wireless NICs and using the PDD removal method, we improve the sensing capability of the off-the-shelf Wi-Fi devices significantly. This provides the ability to extract an additional useful measurement parameter which is the raw phase in Wi-Fi CSI-based sensing system. Using this system, we utilize signal processing methods that have not been possible to implement in previous research with earlier systems and enable it to estimate the direction of a moving target directly from CSI. It can sense both small and large body movements and the results are promising. Additionally, we built the system with a Raspberry Pi to make it low-cost and portable. Further applications like activity recognition and localization can be built based on our system. Moreover, better synchronization methods like wireless synchronization can help a lot to deploy the system in real world and fully reuse the existing communication system for sensing.

VI. Acknowledgment

The authors would also like to thank the researchers of the EPSRC funded OPERA project (Opportunistic Passive Radar for Non-Cooperative Contextual Sensing) for useful discussions and input over the course of the work.

References

[1] B. Tan, Q. Chen, K. Chetty, K. Woodbridge, W. Li, and R. Piechocki, "Exploiting WiFi Channel State Information for Residential Healthcare Informatics," *IEEE Communications Magazine*, vol. 56, pp. 130–137, may 2018.

[2] D. Zhang, Y. Hu, Y. Chen, and B. Zeng, "BreathTrack: Tracking indoor human breath status via commodity WiFi," *IEEE Internet of Things Journal*, vol. 6, pp. 3899–3911, apr 2019.

[3] Y. Gu, X. Zhang, Z. Liu, and F. Ren, "WiFi-based real-time breathing and heart rate monitoring during sleep," *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings*, dec 2019.

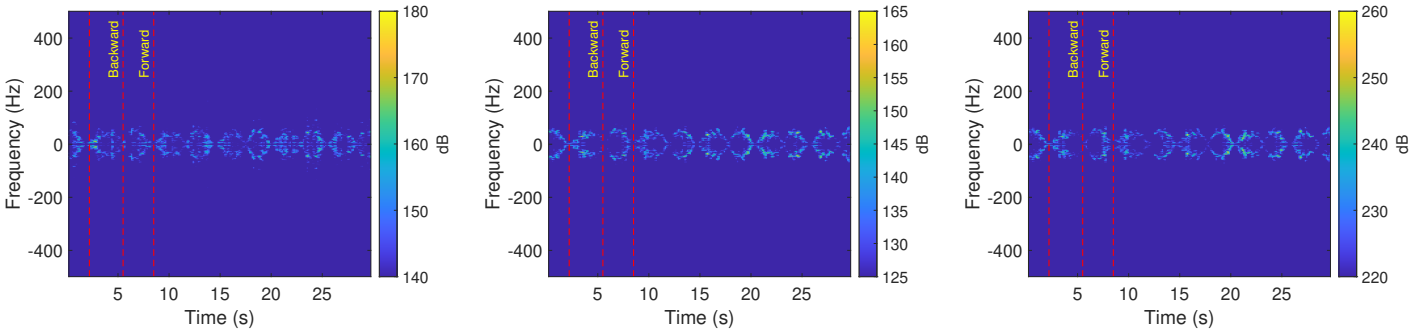
[4] C. Tang, W. Li, S. Vishwakarma, F. Shi, S. Julier, and K. Chetty, "FMNet: Latent Feature-wise Mapping Network for Cleaning up Noisy Micro-Doppler Spectrogram," *IEEE Transactions on Geoscience and Remote Sensing (2021) (In press).*, 2021.

[5] W. Li, R. J. Piechocki, K. Woodbridge, C. Tang, and K. Chetty, "Passive WiFi Radar for Human Sensing Using a Stand-Alone Access Point," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, pp. 1986–1998, mar 2021.

[6] L. Wang, K. Sun, H. Dai, A. X. Liu, and X. Wang, "WiTrace: Centimeter-level passive gesture tracking using WiFi signals," *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON 2018*, pp. 1–9, jun 2018.

[7] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with Wi-Fi," *MobiSys 2019 - Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 313–325, jun 2019.

[8] Q. Chen, B. Tan, K. Woodbridge, and K. Chetty, "Indoor target tracking using high doppler resolution passive Wi-Fi radar," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2015-Augus, pp. 5565–5569, aug 2015.

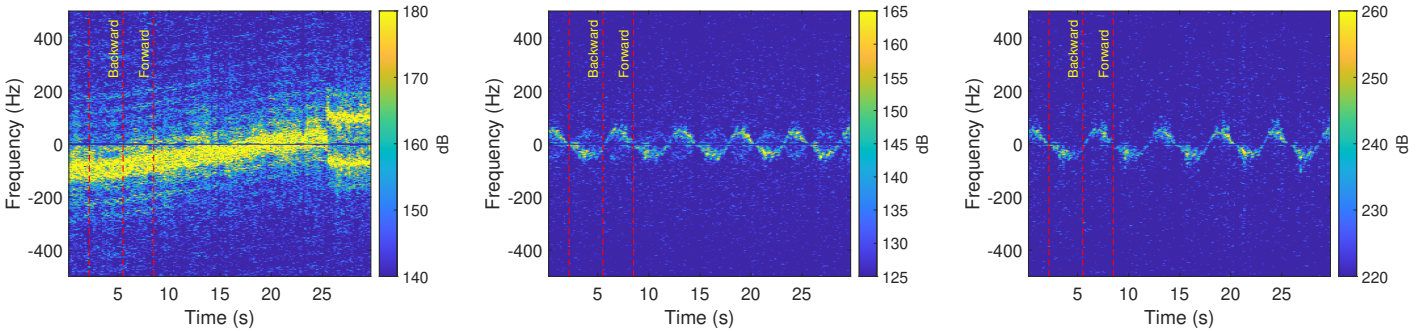


(a) CFO removal: Off, PDD removal: Off

(b) CFO removal: On, PDD removal: Off

(c) CFO removal: On, PDD removal: On

Fig. 8: STFT of CSI amplitude for walking target



(a) CFO removal: Off, PDD removal: Off

(b) CFO removal: On, PDD removal: Off

(c) CFO removal: On, PDD removal: On

Fig. 9: STFT of raw CSI (amplitude and phase) for walking target

[9] K. Qian, C. Wu, Y. Zhang, G. Zhang, Z. Yang, and Y. Liu, "Widar2.0: Passive human tracking with a single Wi-Fi link," *MobiSys 2018 - Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services*, pp. 350–361, jun 2018.

[10] KotaruManikanta, JoshiKiran, BharadiaDinesh, and KattiSachin, "SpotFi," *ACM SIGCOMM Computer Communication Review*, vol. 45, pp. 269–282, aug 2015.

[11] Q. Chen, Y. Liu, F. Fioranelli, M. Ritchie, B. Tan, and K. Chetty, "DopNet: A Deep Convolutional Neural Network to Recognize Armed and Unarmed Human Targets," *IEEE Sensors Journal*, vol. 19, pp. 4160–4172, jun 2019.

[12] K. Chetty, G. E. Smith, and K. Woodbridge, "Through-the-wall sensing of personnel using passive bistatic wifi radar at standoff distances," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, pp. 1218–1226, apr 2012.

[13] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *Computer Communication Review*, vol. 41, p. 53, jan 2011.

[14] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, vol. 2015-Septe, pp. 53–64, sep 2015.

[15] M. Schulz, D. Wegemer, and M. Hollick, "Nexmon: The C-based Firmware Patching Framework," 2017.

[16] IEEE, "802.11-2020 - IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) S," 2021.

[17] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, vol. 2015-Septe, pp. 65–76, sep 2015.

[18] L. Nathawad, M. Zargari, H. Samavati, S. Mehta, A. Kheirkhahi, P. Chen, K. Gong, B. Vakili-Amini, J. Hwang, M. Chen, M. Terrovitis, B. Kaczynski, S. Limotyrakis, M. Mack, H. Gan, M. Lee, S. Abdollahi-Alibeik, B. Baytekin, K. Onodera, S. Mendis, A. Chang, S. Jen, D. Su, and B. Wooley, "A dual-band CMOS MIMO radio SoC for IEEE 802.11n wireless LAN," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 51, pp. 358–360, 2008.

[19] H. Zhu, Y. Zhuo, Q. Liu, and S. Chang, "π-Splicer: Perceiving Accurate CSI Phases with Commodity WiFi Devices," *IEEE Transactions on Mobile Computing*, vol. 17, pp. 2155–2165, sep 2018.

[20] K. Qian, C. Wu, Z. Yang, Y. Liu, and K. Jamieson, "Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi," *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, vol. Part F1291, jul 2017.

[21] S. Tewes and A. Sezgin, "WS-WiFi: Wired Synchronization for CSI Extraction on COTS-WiFi-Transceivers," *IEEE Internet of Things Journal*, vol. 8, pp. 9099–9108, jun 2021.

[22] K. Qian, C. Wu, Z. Yang, C. Yang, and Y. Liu, "Decimeter level passive tracking with WiFi," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, pp. 44–48, oct 2016.