mmRipple: Communicating with mmWave Radars through Smartphone Vibration

Kaiyan Cui[1,2], Qiang Yang[1], Yuanqing Zheng[1], Jinsong Han[3]
[1] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[2]School of Software Engineering, Xi'an Jiaotong University, Xi'an, China
[3]School of Cyber Science and Technology, Zhejiang University, Hangzhou, China
{cskcui,csqyang,csyqzheng}@comp.polyu.edu.hk,hanjinsong@zju.edu.cn

ABSTRACT

This paper presents the design and implementation of *mmRipple*, which empowers commodity mmWave radars with the communication capability through smartphone vibrations. In *mmRipple*, a smartphone (transmitter) sends messages by modulating smartphone vibrations, while a mmWave radar (receiver) receives the messages by detecting and decoding the smartphone vibrations with mmWave signals. By doing so, a smartphone user can not only be passively sensed by a mmWave radar, but also actively send messages to the radar using her smartphone without any hardware modifications to either the smartphone or the mmWave radar. *mmRipple* addresses a series of unique technical challenges, including vibration signal generation, tiny vibration sensing, multiple object separation, and movement interference mitigation. We implement and evaluate *mmRipple* using commodity mmWave radars and smartphones in different practical conditions. Experimental results show that *mmRipple* achieves an average vibration pattern recognition accuracy of 98.60% within a 2*m* communication range, and 97.74% within 3*m* on 11 different types of smartphones. The communication range can be further extended up to 5*m* with an accuracy of 91.67% with line-of-sight path. To our best knowledge, *mmRipple* is the first work that allows smartphones to send data to COTS mmWave radars via smartphone vibrations and will enable many new applications such as vibration-based near field communication and pedestrian-to-sensing-infrastructure communication.

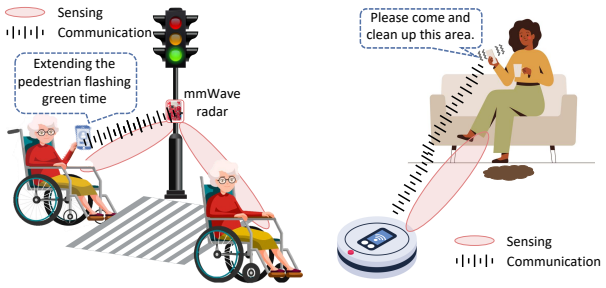
CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

ACM Reference Format:
Kaiyan Cui[1,2], Qiang Yang[1], Yuanqing Zheng[1], Jinsong Han[3]. 2023. mmRipple: Communicating with mmWave Radars through Smartphone Vibration. In *The 22nd International Conference on Information Processing in Sensor Networks (IPSN '23), May 09–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3583120.3586956

IPSN '23, May 09–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0118-4/23/05…$15.00
https://doi.org/10.1145/3583120.3586956



(a) P2I communication (b) Location-based services

Figure 1: Application scenarios of *mmRipple*.

1 INTRODUCTION

Integrated Sensing and Communication (ISAC) is envisioned as a key part of future wireless networks (*e.g.*, beyond 5G and 6G), which effectively utilizes the same wireless spectrum and hardware components to support sensing and communication. Traditional communication technologies (*e.g.*, RFID, Wi-Fi, and LoRa) have been re-purposed for a variety of sensing tasks [5, 11, 20, 26, 32, 57, 58], while radars mainly designed for wireless sensing are also augmented with communication capabilities. Among them, millimeter wave (mmWave) radars have been widely deployed because of i) unprecedented sensing resolution, and ii) robustness to various weather conditions and poor lighting conditions.

These sensing-oriented radars are empowered with communication capabilities mainly by integrating additional communication modules or generating dual-functional waveforms [6, 24, 59]. For example, recent work [14] uses radar's main lobe for sensing, while its side lobes can send data by modulating its transmitting power. In addition, other works [4, 40] use reconfigurable intelligent surfaces (RIS) to communicate with mmWave radars. For example, ROS [34] designs a passive mmWave tag whose layout can be configured to embed data, thus conveying richer information to modern vehicles with mmWave radars and enabling infrastructure-to-vehicle (I2V) communication. These solutions, however, require hardware extension, dedicated mmWave tags, or significant modifications to COTS radars. Therefore, a natural question arises: can we empower COTS mmWave radars with the communication capability for ISAC without any hardware modifications?

In this paper, we present *mmRipple* which builds a direct communication channel from smartphones to mmWave radars without any hardware modifications to either mmWave radars or smartphones. Specifically, *mmRipple* leverages a vibration motor (widely available in almost all smartphones) as a transmitter. The vibra-motor vibrates according to user-defined messages, while the mmWave

Table 1: Comparison with related works.

Type	Technologies & Works	Tx-Rx	Non-contact	No Hardware Modification	No Link Establishment	Motion Robustness	Multi-object Communication	ISAC
Visual	QR Code	Image to Camera	✓	✓	✓	×	×	×
Wireless	Bluetooth	Bluetooth Tx to Bluetooth Rx	✓	✓	×	✓	✓	✓
	Wi-Fi	Wi-Fi Tx to Wi-Fi Rx	✓	✓	×	✓	✓	✓
Acoustic	Dhwani [33]	Speaker to Microphone	✓	✓	✓	×	×	×
	Chirp [23]	Speaker to Microphone	✓	✓	✓	✓	×	×
Vibration	Ripple II [37]	Vibra-motor to Microphone	×	×	✓	×	×	×
	MotorBeat [46]	DC motor to Microphone	✓	×	×	×	✓	×
	mmRipple	**Vibra-motor to mmWave radar**	✓	✓	✓	✓	✓	✓

radar senses such vibrations and decodes these messages. Moreover, *mmRipple* supports multiple objects to send messages simultaneously to a mmWave radar, and each received message implicitly carries the location context of the corresponding object thanks to the spatial information from sensing.

Compared to related works summarized in Table 1, *mmRipple* provides *a contactless communication without any hardware modification or link establishment before direct communication*. Meanwhile, it supports *multiple object communication and is robust to hand and body movements*. To our knowledge, *mmRipple* is the first mmWave ISAC system that enables a mmWave radar to capture smartphone vibration messages while sensing its surroundings.

We envision some applications that could benefit from the communication functionality enabled by *mmRipple*: 1) **Pedestrian-to-infrastructure (P2I) communication**. As shown in Fig. 1(a), pedestrians can leverage their smartphone vibrations to send messages to the traffic light with mmWave radars and actively influence traffic control decisions. For example, after receiving the "crossing" message from an elderly, the traffic light can track her crossing progress to extend the flashing green time for her safety. 2) **Location-based services (LBS)**. Smart appliances equipped with mmWave radars can provide personalized services based on the location of the sender. As shown in Fig. 1(b), the sweeping robot can locate the user and clean the area around her upon request. 3) **Multiple object interaction**. *mmRipple* allows mmWave radars to track multiple users and simultaneously collect their messages in a contact-free manner, which can be used for ticket checking (*e.g.*, COVID-19 vaccine passport checking) to reduce the risk of the virus spreading and shorten the queuing time.

Multiple practical challenges need to be addressed to communicate with mmWave radars through smartphone vibrations. On the transmitter side, a smartphone needs to generate vibration patterns that can be detected and decoded by a mmWave radar. When the vibra-motors of multiple smartphones vibrate concurrently, the reflection signals interfere at the receiver, making it challenging to separate and decode the concurrent vibrations. Furthermore, as a smartphone can be held in hand during communication, hand and body movements may distort smartphone vibrations.

We address the above challenges and develop *mmRipple*. On the transmitter side, we conduct experimental studies to understand smartphone vibrations and design orthogonal vibration patterns that can be easily separable from each other and reliably detected by mmWave radar. On the receiver side, to support multi-object communication, *mmRipple* leverages the spatial diversity of multiple objects to separate the mixed vibration signals in both range and angle with joint Range-FFT operation and beamforming technique. The diversity of vibration frequency of vibra-motors in smartphones is also employed as a feature to separate multiple objects. Furthermore, we mitigate hand and body movements by tracking and recovering integrated vibration signals along target trajectories.

We evaluate the performance of *mmRipple* in various settings. The experimental results on 11 different types of smartphones show that *mmRipple* achieves the average vibration pattern recognition accuracy of 98.60% within a $2m$ communication range, and 97.74% within $3m$. The communication range can be up to $5m$ with an accuracy of 91.67%, when the smartphone is equipped with a Z-axis vibra-motor and has the line-of-sight path to the mmWave radars. The main contributions are summarized as:

- *mmRipple* builds a communication channel from a smartphone to a mmWave radar through smartphone vibrations, without any hardware modifications. To our knowledge, it is the first work that allows COTS mmWave radars to receive smartphone messages, empowering mmWave radars with the communication capability.
- *mmRipple* builds on prior works and makes new scientific contributions by developing novel solutions to address practical challenges: i) understanding and modulating smartphone vibrations; ii) separating multiple objects in mixed reflected signals for multi-object communication; and iii) mitigating movement interference for mobile objects.
- We build a prototype of *mmRipple* using a commodity mmWave radar and different types of smartphones. Comprehensive experiments and evaluation results demonstrate the effectiveness and robustness of *mmRipple*.

2 BACKGROUND

2.1 Vibration Motor

Vibration motors (vibra-motors) are widely used for haptic feedback in smartphones, tablets, and game controllers. Among these vibra-motors, linear resonant actuator (LRA) is the most popular as it supports a faster response time. There are two common types of LRA-based vibra-motors depending on their vibration directions, *i.e.*, Z-axis LRA, and X-axis LRA. The magnetic mass of a Z-axis LRA vibra-motor oscillates back and forth along the direction perpendicular to the smartphone screen. In contrast, an X-axis LRA vibra-motor (*e.g.*, Taptic Engine of Apple) vibrates laterally.

2.2 Primer on mmWave Sensing

Range estimation by Range-FFT. mmWave radar transmits FMCW signals (*i.e.*, chirp signals) to sense objects. The transmitted chirp signal $x(t)$ can be represented as:

$$x(t) = exp[j(2\pi f_c t + \pi K t^2)] \tag{1}$$

where f_c is the starting frequency of the chirp and K represents the chirp slope. After being reflected by the object at distance d, the received signal $y(t)$ can be represented as $y(t) = \alpha x(t - t_d)$, where α is path loss, time delay is $t_d = 2d/c$, and c is the speed of light. Then the mmWave radar mixes $x(t)$ and $y(t)$ and outputs the Intermediate Frequency (IF) signal $s(t)$ which consists of a tone with a beat frequency f_{IF} as [19]:

$$\begin{aligned} s(t) &= x^*(t)\, y(t) \\ &= \alpha \exp[-j(2\pi K t_d t + 2\pi f_c t_d - \pi K t_d^2)] \\ &\approx \alpha \exp[-j(2\pi \underbrace{\frac{K2d}{c}}_{f_{IF}} t + \underbrace{\frac{4\pi d}{\lambda}}_{\phi})] \end{aligned} \tag{2}$$

where λ is the wavelength. We notice that the value of beat frequency f_{IF} contains the distance information. Hence, we can determine the beat frequency f_{IF} by taking FFT (Range-FFT) on the received IF signal $s(t)$, and then the distance d between the object and the radar can be calculated by $cf_{IF}/2K$.

Micro-displacement measurement based on phase changes. The range resolution of radar d_{res} is limited by the chirp bandwidth B, *i.e.*, $d_{res} = c/2B$. For example, $d_{res} = 3.75cm$ when $B = 4GHz$. It means a micro displacement (<3.75cm) will not cause detectable changes in the beat frequency f_{IF}, *i.e.*, peak shift in FFT bins. Fortunately, such a subtle change can still be captured in the phase value ϕ. In Eq. 2, the phase value of IF signal $\phi(t)$ is $4\pi d/\lambda$. If the object distance changes by Δd, the phase value will change accordingly. Hence, Δd can be derived from phase change $\Delta\phi$, *i.e.*, $\Delta d = \lambda\Delta\phi/4\pi$. Hence, mmWave radars can track the phase changes in chirps to capture the micro-displacement above $10\mu m$ [19].

3 SMARTPHONE VIBRATION

3.1 Modelling Smartphone Vibration

To intuitively understand how to recover the smartphone vibration with a mmWave radar, we first show a basic vibration model in Fig. 2. We set the mmWave radar as the coordinate system's origin. The initial location of the vibration source (smartphone) is $S_0(x_0, z_0)$ with the initial range of R_0 to the mmWave radar. When the smartphone vibrates, it will follow a typical harmonic motion

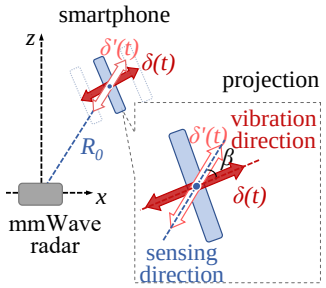


Figure 2: Model.

and produce a time-varying micro-displacement $\delta(t)$ as:

$$\delta(t) = A\cos(2\pi f_v t) \tag{3}$$

where A is the vibration amplitude and f_v is the vibration frequency.

Due to the misalignment between the smartphone's vibration direction and the mmWave radar's sensing direction with angle β, the measured displacement denoted as $\delta'(t)$ is a projection along the sensing direction, *i.e.*, $\delta'(t) = \cos\beta \cdot \delta(t)$. The smartphone range $R(t)$ sensed by the mmWave radar is $R_0 + \delta'(t)$. If we rewrite the object distance d as the smartphone range $R(t)$ in Eq. 2, the received IF signals from the smartphone $s(t)$ can be represented as:

$$s(t) = \alpha exp[-j(\frac{4\pi K}{c}R(t)t + \frac{4\pi}{\lambda}R(t))] \tag{4}$$

Then we extract the reflected signal $S_r(t)$ from this target range by performing a Range-FFT on the IF signal as:

$$s(t) \xrightarrow[\text{in object range bin}]{\text{Range-FFT}} S_r(t) = \alpha \exp[-j\frac{4\pi}{\lambda}R(t)] \tag{5}$$

The corresponding phase measurements from the target range bin can be represented as:

$$\phi_r(t) = \frac{4\pi}{\lambda}(R_0 + \cos\beta \cdot \delta(t))\ mod\ 2\pi \tag{6}$$

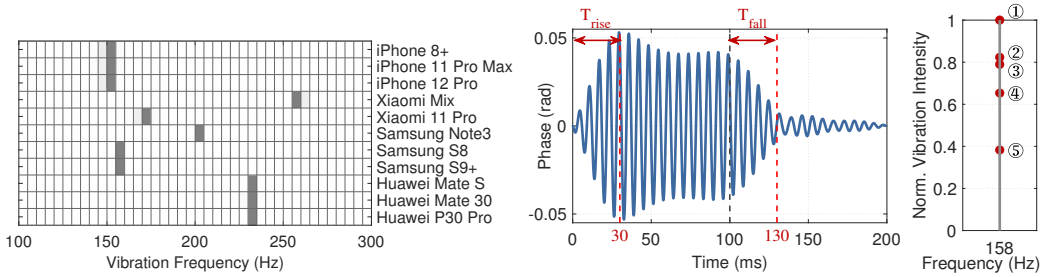
Thus, phase measurements $\phi_r(t)$ can reflect smartphone vibrations.

3.2 Characterizing Smartphone Vibration

We conduct empirical studies to investigate the characteristics of smartphone vibrations. We select 11 popular smartphones and use a mmWave radar (TI AWR1642) to capture their vibrations. In each experiment, one smartphone is placed directly in front of the radar $1m$ away, and its vibrating direction is well-aligned with the mmWave sensing direction, *i.e.*, $\beta = 0$. The ground truth is measured by a piezoelectric vibration meter [2].

Vibration frequency. Fig. 3(a) shows the distribution of the vibration frequencies of the 11 smartphones detected by the mmWave radar. Compared to the ground truth, the measurement error is within $5Hz$, indicating that the mmWave radar can accurately capture smartphone vibrations. In addition, we have two observations: (1) Each smartphone vibrates in a narrow frequency band. This is because vibra-motors are usually set to vibrate around the resonant frequency to produce a better vibration performance [38]. Therefore, we can assume that the smartphone vibration is limited to a single frequency. (2) We also observed that smartphones from different vendors have various vibration frequencies. iPhones vibrate at around 150Hz, while Huawei smartphones vibrate at around 230Hz. Although there are slight differences, the vibration frequencies are generally in the range between $100Hz$ and $300Hz$.

Vibration amplitude and inertia. The peak-to-peak vibration amplitude of the smartphone vibra-motor is typically around $9\mu m$~$248\mu m$, resulting in $0.029rads$~$0.8rads$ phase change for the



(a) Vibration frequency (b) Vibration amplitude and inertia (c) Placement

Figure 3: The characteristics of smartphone vibration.

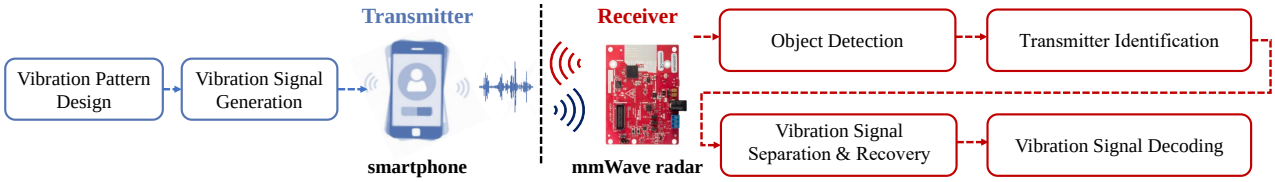


Figure 4: Overview of *mmRipple*.

77GHz mmWave radar. We select Samsung S9+ and let its vibra-motor vibrate for 100ms with the maximum amplitude and stop. Fig. 3(b) plots the vibration signal collected by a mmWave radar. The phase measurements exhibit sinusoidal patterns with larger variance when the smartphone vibrates, while the pattern disappears when the vibration pauses. The measured maximum peak-to-peak amplitude is about 34.1μm (0.11$rads$). The result is accurate and acceptable, which deviates from the ground truth by 3μm. In addition, we observe that the vibra-motor takes a rise time T_{rise} to reach its maximum vibration amplitude due to inertia. It also requires a fall time T_{fall} to stop vibration. Typically, the rise/fall time of smartphone vibra-motors is between 4ms and 120ms.

Vibration effect under different placements. The vibrating smartphone in different placement conditions may produce different vibration effects. Hence, we use the piezoelectric vibration meter (ground truth) that directly touches the smartphone (Samsung S9+) to measure its vibrations in five conditions: (1) on the table, (2) on a tripod, (3) held with fingers, (4) held in the whole palm, and (5) in a pocket. Fig. 3(c) shows the measurement results. Although some additional vibration echos are introduced [7, 17], we can see the measured vibration frequency is not drifted under different placements. In contrast, the measured vibration intensity [1] is indeed affected as the softer contact surface will absorb more vibrations. We also noticed that the vibration effect on a tripod is similar to that held with fingers. Hence, we put a smartphone on a tripod to simulate user habits under laboratory conditions.

4 TRANSMITTER DESIGN

Fig. 4 illustrates the overview of *mmRipple*. In this section, we first elaborate on the design of the transmitter side.

4.1 Vibration Pattern Design

In *mmRipple*, different smartphone vibration patterns are designed to send different messages. Therefore, we need to carefully design the vibration patterns with the following design considerations: i) *High separability.* Designed vibration patterns should be easily separable from each other to reduce recognition errors. ii) *High reliability.* Designed vibration patterns should be accurately and reliably detected by mmWave radars under background noise and interference. To this end, we design a set of patterns with inter-pattern orthogonality and intra-pattern repetition:

Inter-pattern orthogonality means that different vibration patterns are orthogonal to each other, *i.e.*, the cross-correlation between any two patterns is minimized, so that we can differentiate them and reduce matching errors. In particular, we adopt Walsh codes [15], which are widely used in CDMA systems, to guide the pattern design. Table 2 illustrates a set of 4-bit Walsh codes [2]. Each code can be used to generate one pattern.

Table 2: 4-bit Walsh codes

Index	Walsh sequences	Index	Walsh sequences
W_1^4	1111	W_3^4	1100
W_2^4	1010	W_4^4	1001

Intra-pattern repetition requires that a vibration pattern will be repeatedly transmitted several times, thereby improving communication reliability. In this case, one transmission will send K consecutive and identical vibration patterns ($K \geq 1$) to form a pattern frame. Fig. 5 shows a pattern frame, which consists of two consecutive and identical vibration patterns based on the Walsh code W_2^4 (1010). A unique delimiter is used to separate two consecutive patterns and avoid the ambiguity introduced by the intra-pattern repetition. Specifically, we adopt the on-off keying (OOK) to modulate the pattern frame. One symbol duration T_{sym} is divided into 2 smaller time slots T_{slot}. Data-1, data-0, and the delimiter are encoded into vibration modes "on-on", "off-off", and "off-on-off", respectively. Therefore, we can leverage multiple orthogonal codes to design different vibration patterns.

4.2 Vibration Signal Generation

The next issue is how to generate the expected vibration signals. Current smartphones provide APIs [3] to control vibra-motors by

[1] Vibration intensity is measured in m^2/s, which is proportional to vibration displacement.

[2] *mmRipple* uses the inverse-ordered Walsh code.

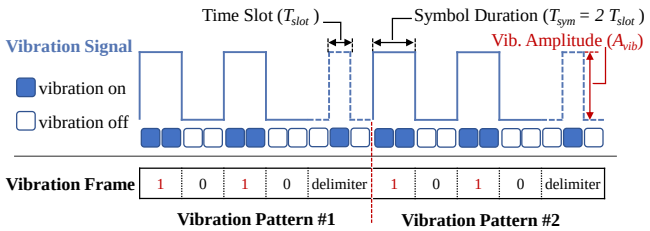


Figure 5: A vibration frame containing two consecutive and identical vibration patterns based on the Walsh code W_2^4.

vibration time T_{vib} in milliseconds and vibration amplitude A_{vib} in the range $[0, 255]$. Therefore, we can use a series of duration and amplitude pairs $\{\langle T_{vib}, A_{vib}\rangle\}$ to control the vibra-motor.

We specify the minimum unit of vibration duration T_{slot} should be greater than or equal to the rise time T_{rise} of the vibra-motor to overcome inertia and reduce inter-symbol interference. For the vibration amplitude, we use $A_{vib} = 255$ to encode the "on" state, and $A_{vib} = 0$ to encode the "off" state. As such, in the vibration "on" state, the motor can vibrate at its highest amplitude in a fully activated state. Conversely, the motor dampens in the vibration "off" state. For example, the vibration pattern in Fig. 5 can be represented by the sequences $\{\langle 60, 255\rangle, \langle 60, 0\rangle, \cdots, \langle 30, 0\rangle, \langle 30, 255\rangle, \langle 30, 0\rangle\}$ when $T_{slot} = 30ms$.

5 RECEIVER DESIGN

In this section, we present the technical details of the receiver side.

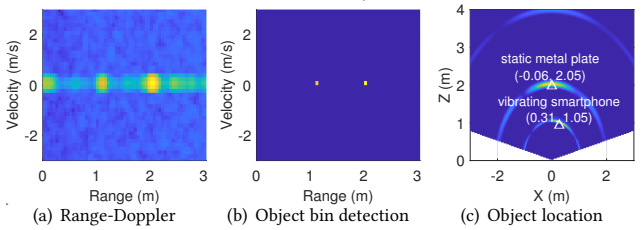
5.1 Object Detection

On the receiver, we first detect surrounding objects, including both transmitters (*i.e.*, vibrating smartphones) and other co-existing objects. To elaborate our system, we conduct a feasibility study in an office, where a mmWave radar is placed at location $(0, 0)$ to sense two objects: a static metal plate at $(0m, 2m)$ and a vibrating smartphone at $(0.3m, 1m)$.

Step 1: Range-Doppler spectrum acquisition. We first perform the Range-FFT on the received ADC samples to detect objects in range and then perform a second FFT (Doppler-FFT) on all chirps in a frame to separate objects in the rate of phase change, *i.e.*, velocity [19]. Hence, we obtain a Range-Doppler spectrum as shown in Fig. 6(a). We observe that there are two bright spots around $1m$ and $2m$ with a velocity of 0, indicating that the objects (*i.e.*, vibrating smartphone and static metal plate) exist at the corresponding range and velocity. Note that although the smartphone is a vibrating object, its vibration velocity is smaller than the velocity resolution (about $0.15m/s$). Thus, its velocity is reported as 0 in the spectrum.

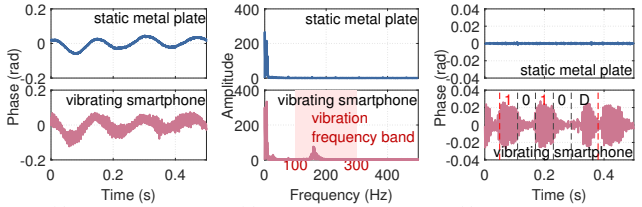
Step 2: CFAR-based object bin detection. This step exploits a constant false alarm rate (CFAR) detection algorithm to search for the bright spots (*i.e.*, candidate objects) in the Range-Doppler spectrum based on an adaptive threshold. Once the magnitude of a bin exceeds a threshold, there can be a candidate object in this bin. We further leverage a moving window to merge neighbor peaks in candidate object detection. As shown in Fig. 6(b), we detect 2 candidates from the Range-Doppler spectrum in Fig. 6(a).

Step 3: Object location extraction. To get the exact object location, we exploit the phase difference of received signals at multiple antennas to calculate the angle of arrival (AoA). Specifically, the distance difference from the object to multiple antennas results



(a) Range-Doppler (b) Object bin detection (c) Object location

Figure 6: Illustration of object detection.



(a) Raw phase (b) Freq. distribution (c) Vibration signal

Figure 7: Illustration of transmitter identification.

in a phase change ω, which is related to the AoA θ [12, 18], *i.e.*, $\theta = arcsin(\lambda\omega/2\pi d_A)$, where d_A is the antenna spacing. Suppose an object is in range bin r and its AoA related to the mmWave radar is θ, the object location (x, z) can be calculated as: $x = r\,sin(\theta)$ and $z = r\,cos(\theta)$. Fig. 6(c) demonstrates the result of object location extraction. We can see that the difference between calculated object location and real object location is negligible ($\leq 6cm$), meaning that mmWave radars can accurately locate the candidate objects.

5.2 Transmitter Identification

After detecting candidate objects, we analyze the reflected signals from each candidate object to detect whether a candidate object is a real transmitter with defined vibration patterns (*i.e.*, a vibrating smartphone) or a static object in the environment.

Fig. 7(a) plots the raw phase measurements of the two objects in Fig. 6(c). Compared to the static object, the vibrating object has a larger variance in phase values. However, phase changes hardly show clear vibration patterns due to the impact of background noises. In the frequency domain (Fig. 7(b)), we observe that both objects suffer from the impact of the Direct Constant (DC) component and low-frequency noises. Unlike static objects, the frequency spectrum of the vibrating object exhibits a sharp peak corresponding to the smartphone vibration frequency range of $100Hz$ to $300Hz$.

Therefore, we can leverage a threshold to identify the vibration objects (transmitters) that vibrate in the vibration frequency band. Once the average amplitude of frequency components in this band exceeds the threshold, the object is detected as a vibration target. We empirically set the threshold as the mean value of noise plus three standard deviations of noise (*i.e.*, 99.7% confidence level).

5.3 Vibration Signal Separation and Recovery

After locating the vibration target, we can leverage a band pass filter (BPF) to recover the expected vibration signal while filtering out noises. Based on our characterization study, we empirically set the lower and the upper stopping frequencies as $100Hz$ and $300Hz$, respectively, which cover the vibration frequencies of most smartphones. Fig. 7(c) plots the vibration signal extracted from the reflection signals. We observe that the recovered vibration signals from the vibrating smartphone clearly exhibit the transmitted vibration pattern, *i.e.*, "1010" followed with a delimiter (D) in Fig. 5.

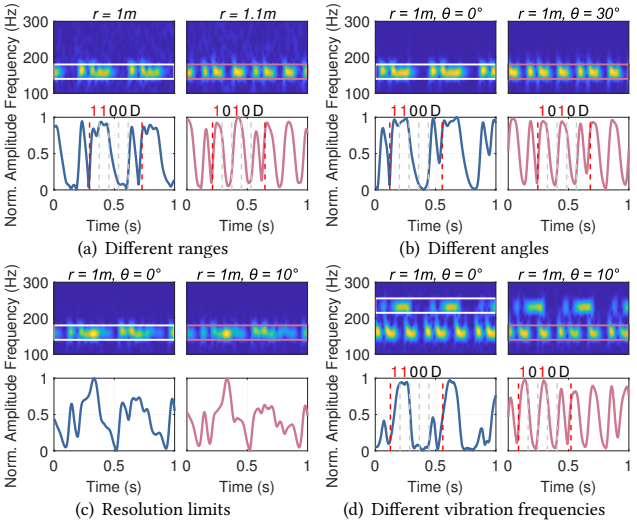


Figure 8: Illustration of multi-object communication.

In practice, multiple smartphones may communicate with a mmWave radar concurrently, leading to interference and collision at the receiver. To tackle this problem, *mmRipple* separates multiple objects in three dimensions: range, angle, and frequency.

Separating multiple objects in range. When multiple objects are vibrating simultaneously in different ranges and each range has only one target, we can directly separate them by the Range-FFT (Eq. 5). Suppose there is a vibrating object in range bin r. Based on Eq. 6, its phase measurements $\phi_r(t)$ extracted from the target range bin r can be represented as:

$$\phi_r(t) = \frac{4\pi}{\lambda}(R_0 + \cos\beta \cdot \delta(t)) = \underbrace{\frac{4\pi}{\lambda}R_0}_{\text{DC}} + \underbrace{\frac{4\pi}{\lambda}\cos\beta \cdot A\cos(2\pi f_v t)}_{\text{vibration signal Y(t)}} \tag{7}$$

The *mod* operation can be ignored, since smartphone vibration is generally very small. We name the second term as *vibration signal* $Y(t)$ and exploit a BPF to extract it, *i.e.*, $Y(t) = BPF[\phi_r(t)]$. Therefore, we can extract the phase measurements of each transmitter from its corresponding range to separate the mixed signals.

We place two vibrating smartphones (Samsung S8 and S9+) in different ranges and capture their vibrations. Fig. 8(a) shows the vibration spectrograms and vibration levels extracted from different ranges. The vibration spectrogram is generated from the vibration signals by Short-Time Fourier Transform (STFT). The vibration level is the normalized energy profile of the target frequency band. When the vibration level is higher, the vibra-motor is in the vibration "on" state; otherwise, it is in the "off" state. We observe from Fig. 8(a) that the vibration signals from different ranges show different vibration patterns, demonstrating that the objects can be separated in range. Note that the range resolution d_{res} will limit the ability to resolve multiple objects in range, which is determined by chirp bandwidth B, *i.e.*, $d_{res} = c/2B$ where c is light speed. In our setting, the range resolution is about $5cm$. This means that *mmRipple* cannot separate two objects if their range spacing is less than $5cm$.

Separating multiple objects in angle. Fortunately, when two vibrating objects locate in the same range, we can exploit the beamforming technique to spotlight on each target. For the object in range bin r and direction θ, we perform beamforming on the reflected signals $S_r^k(t)$ from the object range across all K Rx antennas:

$$S_{r,\theta}(t) = \sum_{k=1}^{K} S_r^k(t) \cdot \exp[-j(k-1)2\pi d_A \sin(\theta)/\lambda] \tag{8}$$

where d_A is the antenna spacing. $S_{r,\theta}(t)$ is the reflected signals from direction θ and range r. In this way, we can separate objects in both range and angle, and extract the specific reflected signals $S_{r,\theta}(t)$ for the target located in range r and direction θ. Then, we can extract the phase measurements $\phi_{r,\theta}(t)$ for each target to recover the corresponding vibration signals.

Here, we place the two smartphones in the same range ($1m$) but in different directions (*i.e.*, 0° and 30°). Fig. 8(b) shows that the vibration signals extracted from different beam steering angles present different vibration patterns. Note that beamforming also increases the SNR of vibration signals and mitigates the multi-path effect, by coherently combining desired signals and randomly adding up background reflections across multiple antennas.

Angular resolution θ_{res} depends on the number of receive antenna N, antenna spacing d_A as well as the AoA of object θ, *i.e.*, $\theta_{res} = \lambda/Nd_Acos(\theta)$. In our setting, the angular resolution is about 28.65° in front of the mmWave with 4 Rx antennas and 1 Tx antenna. If two objects are too close in range and angle, it is hard to separate the two objects. Fig. 8(c) shows the vibration signals of two objects located in the same range $1m$ and different directions (0° and 10°). Since their angular spacing is less than the angular resolution, the vibration signals cannot be separated.

Separating multiple objects in vibration frequency. *mmRipple* can exploit the diversity of vibration frequency (Fig. 3(a)) among different smartphones to further separate multiple objects. To verify this, two smartphones (Huawei Mate 30 and Samsung S8) with different vibration frequencies are placed $1m$ away from the radar and their angular spacing is 10°. As shown in Fig. 8(d), the vibration signals extracted from the corresponding vibration frequencies show the expected vibration patterns. Hence, when these two close objects cannot be separated in either range or angle, the vibration frequencies can be used as another dimension to resolve their collisions. The vibration frequency resolution is determined by the window size in STFT. In our setting, the window size is set to $51.2ms$ with a frequency resolution f_{res} of about $2Hz$.

Overall, *mmRipple* can separate multiple objects in range, angle, and vibration frequency. As long as two vibrating objects differ in one dimension ($5m$ in range, 28.5° in angle or $2Hz$ in vibration frequency), their vibration signals can be separated and extracted.

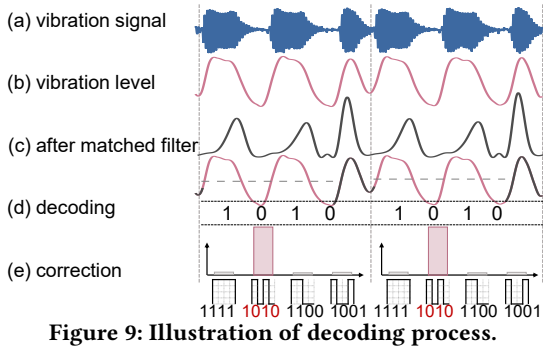
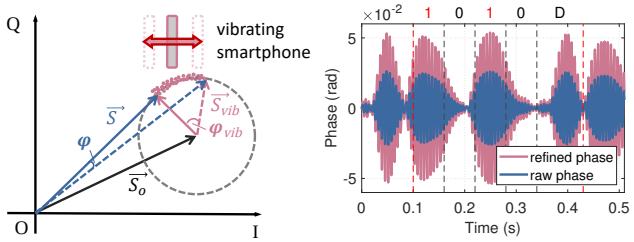


Figure 9: Illustration of decoding process.



(a) Signal model in IQ (b) Remove the static component

Figure 10: Illustration of vibration signal refinement.

5.4 Vibration Signal Decoding

This step aims to decode and recognize the vibration patterns.

(1) Segmentation. We first locate the delimiter to split a consecutive vibration signal into multiple *vibration pattern signals*, each of which is only modulated by one vibration pattern without a delimiter. Unlike data-0 and data-1 which are both represented by two consecutive chips, the delimiter is denoted by three chips "off-on-off". Thus, we exploit a matched filter with the known pattern (a delimiter) to infer the delimiter location. Fig. 9(c) shows the result of applying the matched filter to the recovered vibration level. The highest correlation value reveals the delimiter location. After that, multiple vibration pattern signals are extracted before a delimiter.

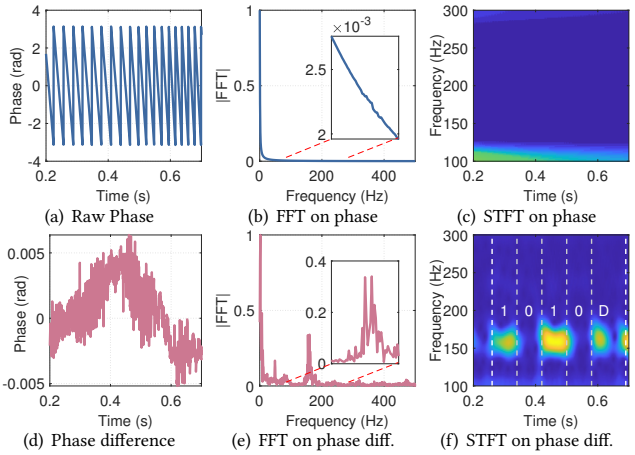
(2) Decoding. Since we adopt the OOK modulation, vibration pattern signals can be directly decoded based on the difference in vibration levels in Fig. 9(d). For each vibration pattern signal, we empirically set the threshold to half the maximum vibration level amplitude of its delimiter. Then in each symbol duration, if the majority of vibration level amplitudes (*e.g.*, 80%) are above the threshold, this symbol is mapped to data-1; otherwise, it is data-0.

(3) Error correction. Due to noise and vibration inertia, the extracted vibrations may have some distortion, causing decoding errors. For example, the vibration pattern "1111" may be incorrectly decoded as a non-defined code "1110". In this case, we will further correct the decoding results using the inter-pattern orthogonality. Specifically, we calculate the cross-correlations to measure the similarity between the extracted vibration pattern levels with the pattern templates in Fig. 9(e). Since our designed vibration patterns are orthogonal, we expect the highest cross-correlation value when the vibration patterns match against themselves. Thus, we leverage the Walsh code corresponding to the maximum cross-correlation value to correct the decoding result and output the final result.

6 SIGNAL REFINEMENT AND MOTION SUPPRESSION

6.1 Signal Refinement

In practice, besides the target vibrating smartphone, other static reflectors (*e.g.*, walls) also reflect signals. Hence, in Fig. 10(a), the received signal from this target $\vec{S}$ is a superposition of the static component $\vec{S}_0$ and the vibrating component $\vec{S}_{vib}$. If the static component is not suppressed, the extracted phase ϕ cannot represent the actual vibration-induced phase changes ϕ_{vib}, degrading the sensing resolution [19]. Therefore, after extracting the reflected signals from the target (Eq. 8), we first perform circle fitting to eliminate the static component, and then extract the ideal vibration-induced phase measurements. In Fig. 10(b), we can see that the vibration signal extracted from the refined phase measurement without the static component presents a better vibration amplitude.



(a) Raw Phase (b) FFT on phase (c) STFT on phase

(d) Phase difference (e) FFT on phase diff. (f) STFT on phase diff.

Figure 11: Illustration of motion suppression.

6.2 Motion Suppression

In practice, a smartphone can be carried by a user in hand. Hand motion and body movement will affect the detection of smartphone vibration signals. We invite a user to move his arm and hand holding the vibrating smartphone towards the mmWave radar from $1m$ to $0.7m$. The smartphone faces towards the radar. The user's location is fixed and his arm moves at a relatively constant speed.

Fig. 11(a) shows the phase values $\phi_{r,\theta}(t)$ from the detected target location where $r = 1m$ and $\theta = 1.79°$. We notice that the smartphone movement results in the phase change from $-\pi$ to π, drowning out the tiny smartphone vibrations. In the frequency domain, such minute vibration signals are also dominated by the movement (Fig. 11(b)). Even after applying a band pass filter, vibration patterns can no longer be observed in the vibration spectrogram (Fig. 11(c)).

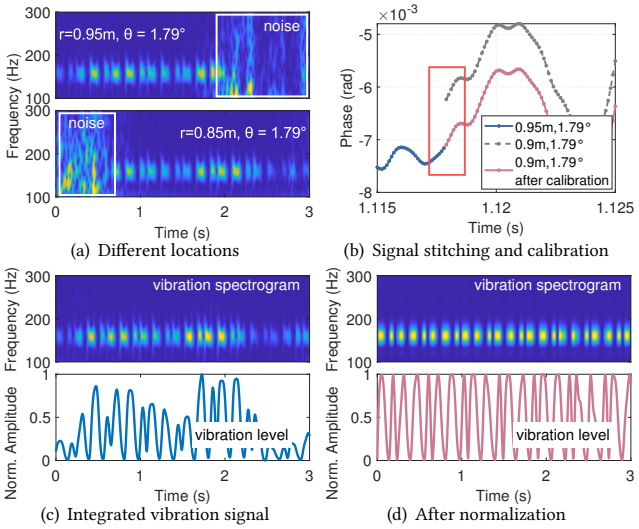
To combat the transmitter motions, we propose a frame-aware motion suppression method. Suppose there is a vibrating smartphone located in (r, θ) with a moving speed v. Considering that the moving speed of transmitter v can be approximated as a constant within a short frame (*e.g.*, $12.8ms$ in our setting), we first remove the static component for each frame through the circle fitting and obtain the refined phase measurement that can be represented as:

$$\begin{aligned}\phi_{r,\theta}(t) &= \frac{4\pi}{\lambda}(R_0 + vt + \cos\beta \cdot \delta(t)) \\ &= \underbrace{\frac{4\pi}{\lambda}R_0}_{\text{DC}} + \underbrace{\frac{4\pi}{\lambda}vt}_{\text{movement}} + \underbrace{\frac{4\pi}{\lambda}\cos\beta \cdot A\cos(2\pi f_v t)}_{\text{vibration signal Y(t)}}\end{aligned} \tag{9}$$

Next, to extract the vibration signal while eliminating the impact of movement as well as DC, we take the first-order derivative of phase measurements $\phi'_{r,\theta}(t)$ for every frame as:

$$\phi'_{r,\theta}(t) = \underbrace{\frac{4\pi}{\lambda}v}_{\text{movement}} \underbrace{-2\pi f_v \cdot \frac{4\pi}{\lambda}\cos\beta \cdot A\sin(2\pi f_v t)}_{\text{vibration signal } Y'(t)} \tag{10}$$

In this way, the DC component is removed and the impact of smartphone movement is transformed into a new DC component. Moreover, the expected vibration signals is amplified by a scaling factor



(a) Different locations (b) Signal stitching and calibration
(c) Integrated vibration signal (d) After normalization

Figure 12: Illustration of vibration signal stitching.

of $2\pi f_v$. Then, we correct the discontinuity at every two consecutive frames [43] and extract the corresponding vibration signal $Y'(t)$ from $\phi'_{r,\theta}(t)$ with a band pass filter.

In practice, we use the phase difference of two consecutive measurements to approximate the first-order derivative of the phase. Fig. 11(d) shows the change of phase difference. Although the phase difference changes are very small and drift because of movement, its frequency spectrum (Fig. 11(e)) exhibits a sharp peak in the smartphone vibration range of $100Hz$~$300Hz$. After applying a band pass filter and STFT, we can observe a clear vibration pattern in the vibration spectrogram (Fig. 11(f)).

6.3 Vibration Stitching along Target Trajectory

In the above discussion, we extract the vibration signal from a specific location of a target. In practice, however, the target location is constantly changing when it is moving, making it impossible to extract the whole vibration signal from one specific location.

Fig. 12(a) exhibits the vibration spectrograms extracted from different locations. We see that the extracted signals from one location only measure the smartphone vibrations when the phone is in that particular location. If the smartphone is not present in that location, the measurements are mainly noise. To handle the movement, we need to track the location of the vibration object and stitch the vibration signals along its moving trajectory.

In a frame, mmWave radar will report a set of points (detected objects in § 5.1). We define the i-th point in j-th frame as $p_{i,j}$, which is a 4D state vector composed of coordinates on x, z axis, velocity v and the reflection intensity ϵ, *i.e.*, $p_{i,j} := [x, z, v, \epsilon] \in \mathbb{R}^4$. Note that the coordinates on y axis will also be reported on advanced radars. Once a vibration target is detected (§ 5.2), we record its current state as the initial state $o_0 := [x_0, z_0, v_0, \epsilon_0]$ and then track it by monitoring the following consecutive frames. Specifically, in frame n, we first calculate the Euclidean distance between the latest target state o_{n-1} and the states of other candidate objects in this frame $p_{i,n}$. Then, the Hungarian algorithm [22, 39] is adopted to find the associated object o_n with a minimum total Euclidean distance in frame n. Meanwhile, we use the Kalman Filter [49, 60] to further

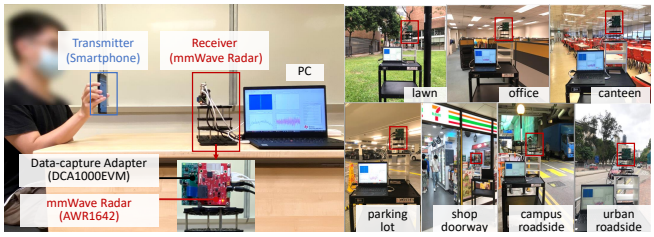


Figure 13: Experiment setting.

predict and correct the target tracks by analyzing the measurements observed over time. Therefore, we can obtain the target state set $O = \{o_0, o_1, ..., o_n, ..., o_{N-1}\}^T \in \mathbb{R}^{N\times 4}$ across N frames and extract its moving trajectory $\{X, Z\} \in O$.

Then we stitch the vibration signal along the target moving trajectory. At each location, we first extract the phase difference measurements for recovering the vibration signals (Eq. 10). Fig. 12(b) shows the phase differences captured at two adjacent locations. Due to different initial phase values and noise, there is a slight gap between the two measurements from adjacent locations, which might cause errors. Hence, we align and stitch the discontinuous phase difference measurements and output the integrated vibration signal along the target moving trajectory. After that, the vibration spectrogram in Fig. 12(c) shows continuous smartphone vibrations. However, there are some distortions in the vibration level due to various multipath effects and noise at different locations. Inspired by peak normalization in audio processing, we adjust the recovered vibration amplitude by normalizing the vibration peak magnitude to a specified level. After normalization (Fig. 12(d)), smartphone vibrations have a stable vibration amplitude that can be decoded.

7 EVALUATION

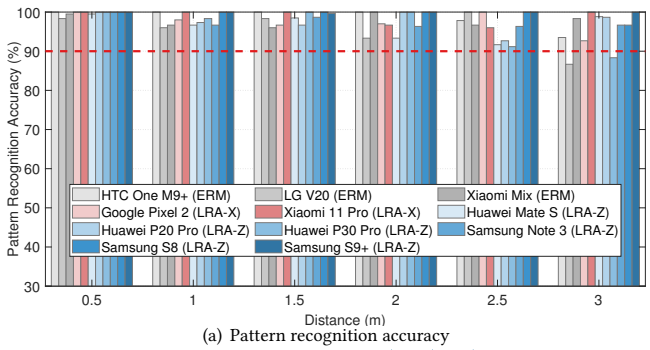
7.1 Experimental Methodology

Hardware and software. As shown in Fig. 13, we implement a prototype of *mmRipple* using commercial off-the-shelf devices. For the transmitter, we test 11 smartphones from 6 vendors equipped with different types of vibra-motors, including traditional eccentric rotating mass (ERM) vibra-motors, X-axis and Z-axis LRA vibra-motors as summarized in Table 3. The receiver is a commercial mmWave radar, Texas Instruments AWR1642, working in the $76GHz$~$81GHz$ frequency band. There are 2 Tx and 4 Rx antennas on the radar board. The ADC samples are captured through a TI DCA1000EVM data acquisition board and then transmitted to a computer with an Intel Core i7-10510U 2.30GHz CPU for processing. Data processing algorithms are implemented in Matlab.

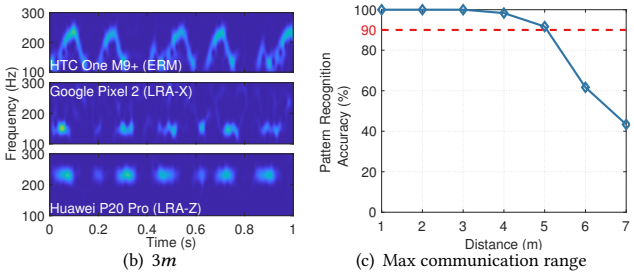
Table 3: Tested smartphones

Vibra-motor Type	Smartphone
ERM vibra-motor	HTC One M9+, LG V20, Xiaomi Mix
X-axis LRA vibra-motor	Google Pixel 2, Xiaomi 11 Pro
Z-axis LRA vibra-motor	Huawei Mate S, P20 Pro, P30 Pro; Samsung Note3, S8, S9+

Experiment setting. On the transmitter side, we control vibra-motors in smartphones to generate vibration patterns with different coding bits, different vibration amplitudes and different time slots. By default, smartphones are fixed on tripod mounts and transmit 4-bit Walsh codes as vibration patterns at maximum vibration amplitude. The vibration time slot is $40ms$ and the vibration direction



(a) Pattern recognition accuracy



(b) $3m$ (c) Max communication range

Figure 14: Overall performance.

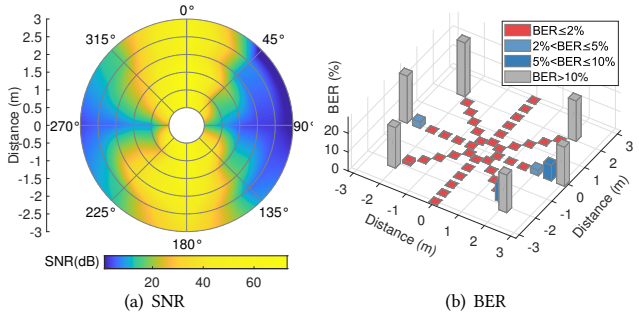
of the smartphone is aligned with the mmWave radar's sensing direction. On the receiver side, we configure the Tx1 of mmWave radar to send FMCW chirps at the starting frequency of $77GHz$ with $3GHz$ bandwidth, and Rx1 ~ Rx4 to receive the reflected signals. The ADC sampling rate of the mmWave radar is $5MHz$ with 256 samples per chirp and the chirp duration is $100\mu s$. Since we only collect one vibration signal sample per chirp, the chirp sampling rate is $10kHz$. The transmitted vibration patterns are recorded as ground truth in the smartphones for performance evaluation.

Metrics. We evaluate *mmRipple* using three metrics: Signal-to-Noise Ratio (SNR), Bit Error Rate (BER), and pattern recognition accuracy. SNR measures the quality of recovered vibration signal, which is defined as the ratio of the strength of vibration signal to that of background noise. BER measures the accuracy of data transmission after error correction. Pattern recognition accuracy is defined as the rate that *mmRipple* correctly matches recovered vibration patterns to their corresponding programmed patterns. Unlike BER which refers to a single bit, it measures if *mmRipple* can correctly recognize a pattern represented with multiple bits.

7.2 Overall Performance

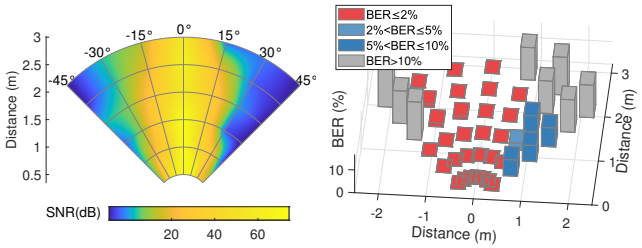
Fig. 14(a) shows the overall performance of *mmRipple* on 11 popular smartphones at different communication distances varying from $0.5m$ to $3m$. To capture better vibrations, the screens of smartphones with EMR or Z-axis LRA vibra-motors (vibrating perpendicular to the screen) face towards the mmWave radar. In contrast, smartphones with X-axis vibra-motors vibrate laterally and we orient their sides towards the mmWave radar. We see that *mmRipple* has high accuracy across different types of vibra-motors in different smartphones. Overall, *mmRipple* achieves the average vibration pattern recognition accuracy of 98.60% within a $2m$ communication range, and 97.74% within $3m$.

When the communication range is greater than $2m$, we observe that recognition accuracy decreases. We compare the vibration signals extracted by three smartphones equipped with different types



(a) SNR (b) BER

Figure 15: Performance centered on the transmitter. The smartphone is fixed at (0,0) and a mmWave radar is placed around it to capture its vibrations.



(a) SNR (b) BER

Figure 16: Performance centered on the receiver. A mmWave radar is fixed at (0,0) and a vibrating smartphone is placed within its FoV.

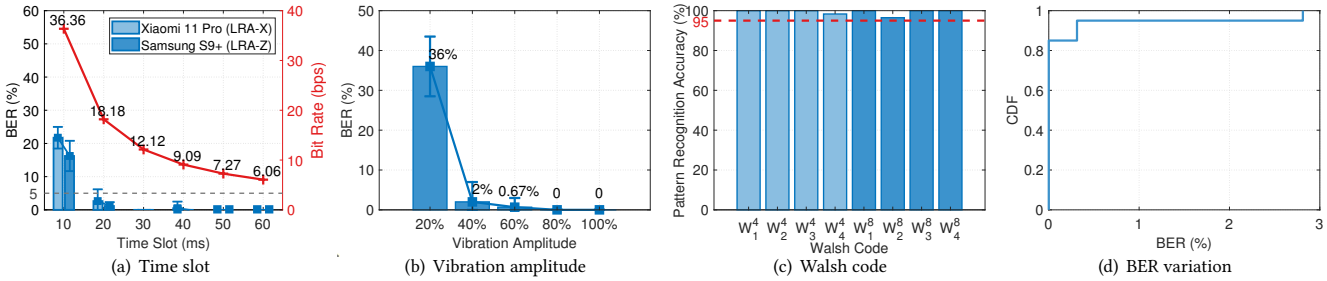
of vibra-motors at $3m$ (Fig. 14(b)) and notice that the vibration signal becomes less prominent as the distance increases. There are two main reasons: 1) weaker reflected signals because of signal attenuation over longer communication range and small form factors of smartphones; 2) more interference and noises from a larger sensing area. In this case, tiny vibration signals are more likely to be drowned by noises and interference. Besides, we also notice that smartphones with LRA vibra-motors perform better than those with traditional ERM vibra-motors. This is because ERM vibra-motors generate vibration with unbalanced mass rotation, which takes a longer rise time to reach the expected vibration amplitude and frequency than LRA vibra-motors.

In addition, we evaluate the system performance at longer distances with Samsung S9+. The vibration direction is well-aligned with the mmWave sensing direction. As shown in Fig. 14(c), we can see that the pattern recognition accuracy decreases with distance. At $5m$, the accuracy can still achieve 91.67%.

7.3 Orientation and Distance

Performance centered on the transmitter. We place the mmWave radar around the fixed smartphone to evaluate the performance centered on the transmitter in Fig. 15. A Samsung S9+ is placed at the origin of coordinates $(0, 0)$. Its screen faces $0°$ and vibration direction is along the Z-axis (the line of $0°$ and $180°$). A mmWave radar captures the smartphone vibrations from different sensing ranges ($0.5m$~$3m$) and directions ($0°$~$315°$).

We have the following key observations: (i) The performance of *mmRipple* decreases as the communication range increases. When the communication range exceeds $2m$, the reflected signal from the smartphone will become weaker, leading to lower SNRs. In this case, it is challenging to accurately discriminate the vibration states,



(a) Time slot (b) Vibration amplitude (c) Walsh code (d) BER variation

Figure 17: Communication performance under different conditions.

leading to high BER (>10%). (ii) The performance of *mmRipple* will degrade with the increase of misalignment between the mmWave radar sensing direction and the smartphone vibration direction. When these two directions are completely aligned (0° and 180°), the captured vibration signal has the largest vibration displacement and thus a higher SNR and lower BER. When the misalignment increases, the captured vibration displacement becomes smaller, resulting in weaker SNRs and higher BERs. If the vibration direction is orthogonal to the sensing direction (90° and 270°), the amplitude of the vibration signal is extremely tiny, but can still be detected due to the distance changes in a near field range.

Overall, *mmRipple* can support $2m$ communication range in any direction, with a BER of 0.893%. Within this communication range, a user can hold the smartphone to send vibration signals in *any direction*. When the communication range is $3m$, the average BER is still 3.4% despite higher BERs in certain directions with misalignment.

Performance centered on the receiver. Next, we fix the radar at $(0, 0)$ and move the smartphone to different angles ($-45°$~$+45°$) and distances ($0.5m$~$3m$). The radar always keeps facing towards 0° and the vibration direction is fixed along the Z-axis. When the phone is at 0°, its vibration direction is perfectly aligned with the sensing direction of the radar.

In Fig. 16, we have the following findings: (i) The performance of *mmRipple* also decreases as the sensing angle increases due to the misalignment between the sensing direction and the vibration direction and the non-uniform antenna radiation pattern. The limitation of radar's field of view (±45°) results in a significant drop in SNR, when the phone gets closer to the boundary of the field. (ii) The performance of *mmRipple* near Tx antenna (left side) is better than the other side. The reason for this might be that the limited size of the smartphone makes its reflected signals susceptible to the non-uniform radiation pattern of antenna and interference [47]. (iii) In the FOV of mmWave radar, it can correctly capture and detect vibration signals from the smartphone, with a BER of 1.88% within $2m$. As the range increases to $3m$, BER increases to 3.78%. Enhanced by the error correction capability with orthogonal coding, the smartphone vibrations in the FOV of mmWave radar and the communication range of $3m$ can be accurately captured by the receiver of *mmRipple*. In practice, we can transmit consecutive and identical vibration patterns and leverage intra-pattern repetition to improve the reliability of communication.

Performance at various height differences. In this experiment, we study the impact of height difference between the smartphone and mmWave radar. A Samsung S9+ and a mmWave radar (AWR1642) are placed at a distance of $2m$ from each other and $1.5m$ off the ground. Then, we increase the height of the smartphone with a step size of $10cm$. Table. 4 shows that *mmRipple* can still achieve a mean BER of 3.8% when the height difference is $50cm$. BER significantly increases when the height difference exceeds $50cm$. This is because it exceeds the limit of the elevation angle of the antenna (±15°, *i.e.*, $53.6cm$ for $2m$) [1], so that the smartphone vibrations cannot be captured. In practice, we can deploy more radars to extend the sensing height.

Table 4: The impact of height difference.

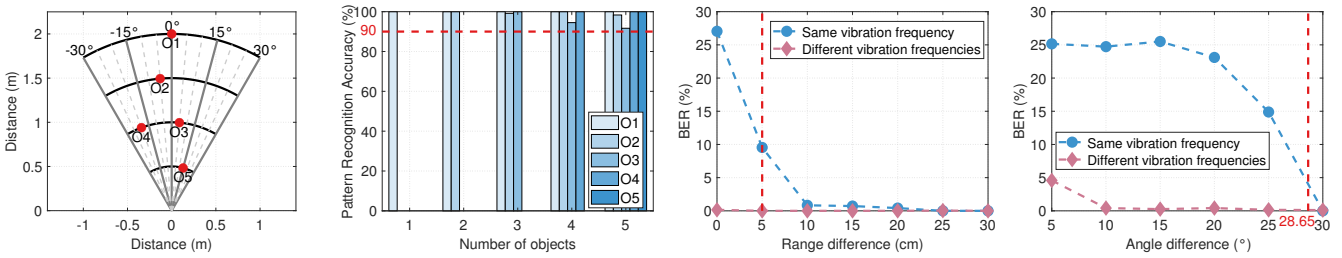
Height diff. (cm)	0	10	20	30	40	50	60
BER (%)	0	0	0	0	3	3.8	11.4

7.4 Communication Performance

In this evaluation, the smartphone and a mmWave radar (AWR1642) are placed $1m$ apart and we test the communication performance at different conditions. By default, the smartphone is Samsung S9+.

Vibration time slot. We vary the vibration time slot from $10ms$ to $60ms$ to evaluate its impact on BER and bit rate on two types of smartphones, *i.e.*, Xiaomi 11 Pro (LRA-X) and Samsung S9+ (LRA-Z). We can see from Fig. 17(a) that BER decreases as the vibration time slot increases, since a longer vibration time slot ensures that the motor has sufficient time to startup and shutdown to counteract the effect of inertia. In this case, vibration states become more prominent and easier to decode. When the vibration time slot is $20ms$, the average BER for these two smartphones is 1.90% and the corresponding bit rate is about $18.18bps$. If the vibration time slot is longer than the vibra-motor's rise time ($40ms$ for Xiaomi 11 Pro and $30ms$ for Samsung S9+), the average BER is less than 1%. As such, we recommend that the vibration time slot should be longer than the smartphone's rise time. On the contrary, a longer vibration time slot will result in a lower bit rate. Therefore, this is essentially a trade-off between decoding accuracy and data rate. In practice, we empirically set the time slot to be $40ms$ to strike a balance, since a short time slot (*e.g.*, $< 40ms$) leads to high decoding errors. We believe such a bit rate is sufficient for *mmRipple* to support various applications like pedestrian-to-infrastructure interaction.

Vibration amplitude. A stronger vibration amplitude means that the displacement change is larger, which yields better recognition performance. In this experiment, we vary the vibration amplitude level (as a percentage of maximum vibration amplitude, *i.e.*, 255) to evaluate its impact. In Fig. 17(b), the results demonstrate that vibration amplitude indeed influences BER. When the vibration amplitude is set higher than 60% of the maximum amplitude, mmRipple can accurately decode vibration signals. To optimize the



(a) Five objects from O1-O5 (b) Accuracy of different number of objects (c) Range difference (d) Angle difference

Figure 18: Multiple object communication.

performance, we set the amplitude to the maximum in our experiments if not specified otherwise. We leave the amplitude adaptation and energy-efficient vibration modulation for future work.

Different Walsh codes. In *mmRipple*, different codes represent different user requests. To investigate the performance of different codes, we select 8 Walsh codes from 4-bit Walsh codes W^4 and 8-bit Walsh codes W^8 to generate different vibration patterns to compare their pattern recognition accuracy. In Fig. 17(c), all Walsh codes show good performance with a pattern recognition accuracy of >96.47%. Hence, any of Walsh codes can be used to deliver a user request and a user can choose the code according to his requirement.

Temporal stability. Given that a vibra-motor is essentially a mechanical component, we next evaluate its temporal stability for the purpose of vibration based communication. We set the vibration time slot to $60ms$ and let a smartphone vibrate at this low bit rate to transmit longer packets for 300 seconds. The experiment is repeated 40 times and the BER is calculated every 15 seconds. We can see from Fig. 17(d) the BER is less than 1% for 95% communication, demonstrating the sufficient temporal stability of the vibra-motor.

7.5 Robustness of mmRipple

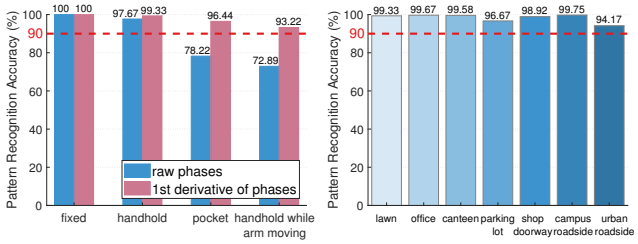
Multiple objects. One appealing characteristic of *mmRipple* is its capability of separating vibration signals of multiple objects and decoding their concurrent messages. As shown in Fig. 18(a), five smartphones from different vendors are fixed on tripod mounts and placed in an area of $2m \times 1m$ in front of the mmWave radar with no obstruction. Among them, object O3 and O4 have the same communication range and different directions. These smartphones transmit different vibration patterns based on 4-bit Walsh codes with $40ms$ vibration time slot.

Fig. 18(b) shows the pattern recognition accuracy of these five objects. As the number of objects increases, the pattern recognition accuracy drops slightly. However, even for five objects, *mmRipple* can still correctly separate multiple vibration objects at different ranges and directions as well as extract the vibration signal for each target independently, achieving a pattern recognition accuracy higher than 91.6%. Moreover, *mmRipple* supports concurrent reception from multiple transmitters, allowing the aggregated throughput to be multiplied. In this experiment, the bit rate of each transmitter is $9.09bps$ with a vibration time slot of $40ms$. On the receiver side, the aggregated throughput for receiving vibration messages from five transmitters is approximately 5×.

Distance between objects. The capacity of *mmRipple* to resolve multiple objects is limited by the sensing resolution. To evaluate its capacity, we place two smartphones (Samsung S8 and S9+) with the same vibration frequency $1m$ away from the radar. We vary the smartphone positions so that they are separated by $0cm$~$30cm$ in range and 5°~30° in AoA relative to the radar. In Fig. 18(c) and Fig. 18(d), we observe that the average BER decreases with the increase of the range and the angle. When the two objects are separated by more than $5cm$ in range or 28.65° in AoA, we can accurately separate the two objects with a BER of less than 1%. We repeat the experiment using two smartphones (Samsung S8 and Huawei Mate 30) with different vibration frequencies. We see that *mmRipple* shows better performance by separating the vibration signals in the frequency domain. Even the two smartphones are very close in range and angle, the average BER is less than 5%, which indicates that the vibration frequency can be used as an additional dimension to further separate objects.

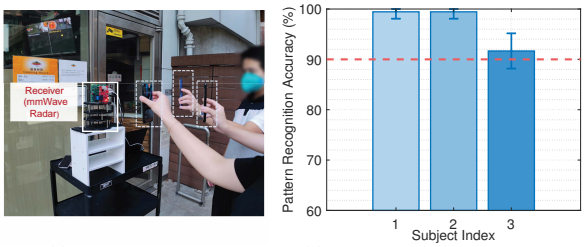
Placement and movement. To evaluate the ability of *mmRipple* to handle objects in different placement and movement scenarios, we conduct experiments in the following four scenarios: (1) smartphones fixed on tripod mounts, (2) handheld smartphones, (3) smartphones in pockets, and (4) handheld smartphones with arm movement. The smartphone faces towards the mmWave radars and the position of the user is fixed in Scenario (2)~(4). In each scenario, we invite three volunteers and collect 150 messages. Fig. 19(a) illustrates the performance of *mmRipple* in these scenarios. Overall, *mmRipple* can effectively mitigate the impact of human movement with a pattern recognition accuracy of >93.22%. When the vibrating phone is in the pocket, we still capture the expected vibrations with an accuracy of 96.44%. In addition, our proposed method based on the first-order derivative of phases outperforms the method of extracting the smartphone vibrations from raw phases. Especially in the scenario of handheld smartphones with arm movement, the pattern recognition accuracy has been improved by 20.33%. *mmRipple* has demonstrated its effectiveness under the impact of hand and arm movement. For larger body movements (*e.g.*, walking and running), the performance of *mmRipple* will decrease as such larger movements will produce a wide range of frequency components spanning a long period of time, easily overwhelming the weak vibration patterns [10]. We leave this for future improvement.

Environmental disturbance. To evaluate the environmental disturbance in practice, we conduct experiments on seven working scenarios in Fig. 13. In each scenario, the distance between the smartphone user and the mmWave radar is kept at $1m$~$1.5m$ and 120 vibration messages are collected. In Fig. 19(b), we see that *mmRipple* has good performance on an open space lawn, as well as in the multipath-rich office, canteen, and parking lot. Even in the scenarios with moving object interference, *e.g.*, near the shop doorway where people frequently pass by, campus and urban roadside with moving vehicles and pedestrians, *mmRipple* achieves a pattern



(a) Placement and movement. (b) Environmental disturbance

Figure 19: Robustness of *mmRipple*.



(a) Practical scenario. (b) Accuracis of different subjects

Figure 20: Case study: multi-object communication.

recognition accuracy higher than 94.17%. The results demonstrate that *mmRipple* can effectively filter out the interference and extract vibration signals by leveraging the prior knowledge of vibration frequencies and vibration patterns.

7.6 Case Study

We take a practical case to present the *mmRipple* performance in a multi-object communication scenario. As shown in Fig. 20(a), a mmWave radar is deployed at the entrance of a canteen to monitor its surrounding objects and meanwhile receive messages from vibrating smartphones. Three volunteers are invited to hold their smartphones (1-Samsung S9+, 2-Samsung S8, 3-Xiaomi Mix) and simultaneously interact with the mmWave radar within a communication range of $2m$. These smartphones send different 4-bit vibration patterns with a vibration time slot of $40ms$. We collect 30 vibration patterns from each experiment and repeated for 5 times. Fig. 20(b) plots the pattern recognition accuracy of these three subjects. We observe that *mmRipple* achieves an average pattern recognition accuracy of 96.85% for three subjects demonstrating its practicality. Subject 3 holding the smartphone with a traditional ERM vibra-motor still has an accuracy of 91.67%. Therefore, we believe that *mmRipple* can be deployed in real-world scenarios to provide both multi-object sensing and multi-object communication, enabling more innovative applications.

8 LIMITATIONS AND DISCUSSION

One-way communication. *mmRipple* only supports one-way communication where a message is sent from a smartphone user to a mmWave radar. In practice, same as a QR code scanner, *mmRipple* can acknowledge to a user via an out-of-band channel (*e.g.*, light, sound, action and visual cues). For example, after receiving a valid message from a user, a door can automatically open and allow access, which implicitly acknowledges the successful reception of the vibration message. Smart traffic lights can similarly acknowledge the user by turning green.

Communication capacity. As the vibration frequency of the vibra-motor in smartphones cannot be configured, *mmRipple* adopts the OOK modulation, where the maximum amplitude vibration represents data-1 and no vibration represents data-0. Higher vibration amplitude can support a longer communication range, while it will also increase the rise time of the vibra-motor due to the inertia, resulting in a lower bit rate. There is a trade-off between communication range and transmission rate. For a vibra-motor with a rise time of $30ms$, its bit rate sending 4-bit patterns is $12.12bps$ at the maximum vibration amplitude. A message (*e.g.*, crossing) takes only $0.33s$. Hence, we believe such a bit rate is sufficient for some applications. For example, in P2I communication, the number and type of requests from pedestrians are limited due to safety concerns and resource allocation in the transportation system. Moreover, *mmRipple* supports concurrent reception, allowing the receiver to collect multiple messages from different transmitters simultaneously.

Communication range and coverage. *mmRipple* can enable a $5m$ communication range when the smartphone vibrates directly in front of the radar with no misalignment between vibration direction and mmWave sensing direction. We believe such a communication range can support short-range applications. In other directions with misalignment and non-line-of-sight scenarios, weak and even no vibrations can be detected. Therefore, we may rotate or deploy more mmWave radars to track the target and extend the communication coverage. Once the response time exceeds a threshold (*e.g.*, $5s$), *mmRipple* can be switched to the intra-pattern repetition mode to send the same message repeatedly, while prompting the user to adjust his interactive gesture and position to avoid misalignment and obstacles. Note that the communication range differs from the sensing range, which is determined by the ADC sampling rate of mmWave radar (about $12m$ in our setting).

Multi-object communication. *mmRipple* separates multiple objects in time, range, angle, and vibration frequency, so that the aggregated throughput on receiver can be multiplied by reading concurrent messages. It is challenging to separate two objects that are almost the same (below resolution) in all dimensions. One possible enhancement is to improve resolution by increasing chirp bandwidth, the number of receive antennas as well as cascading multiple radars [8]. Furthermore, we may exploit differences in the vibration patterns from different transmitters (*e.g.*, orthogonal codes) to enhance the separation capability.

Energy consumption. *mmRipple* uses smartphone vibrations to communicate with mmWave radars. On the receiver side, mmWave radars are typically powered by dedicated power supplies. On the transmitter side, motor vibrations indeed consume the energy of smartphones. In contrast to other smartphone applications, *mmRipple* only requires vibra-motors to work occasionally to complete one communication in a short period of time (*e.g.*, 0.33s). As such, we believe the energy consumption of *mmRipple* is affordable for infrequent usage scenarios.

9 RELATED WORKS

mmWave sensing. Unlike other wireless sensing technologies [9, 16, 27, 44, 45, 48, 52, 53], mmWave sensing can achieve a very high sensing resolution and accuracy [13, 21, 29, 30, 35, 39, 51,

54, 56]. Recent works have also demonstrated its excellent performance in detecting vocal vibration for speech recognition and micro-vibration in industrial scenarios. For example, WaveEar [50] uses mmWave radars to build a noise-resistant speech sensing system for voice-user interaction. Wavoice [28] fuses mmWave signals and audio signals to facilitate accurate speech recognition under complex conditions with noise and motion interference. mmVib [19] measures the amplitude and frequency of tiny vibrations using mmWave radars. Multi-Vib [55] designs a physical marker and leverages mmWave radars to monitor multiple points' vibrations. Unlike these works that measure vibrations of fixed objects, our work *mmRipple* leverages modulated smartphone vibrations to build a communication channel between smartphone users and mmWave radars. Besides, *mmRipple* addresses a few practical challenges, such as human movement, multiple vibrators, *etc.* Recent works [36, 41] have used radars to capture displacements of the water surface caused by underwater acoustic signals, enabling cross-medium communication. However, these works exploit some dedicated devices and focus on communication across the water-air boundary. In contrast, *mmRipple* allows smartphone users to send messages to mmWave radars without any hardware modification.

Vibration based communication. Ripple [38] builds a communication channel, which modulates messages using vibration motors and decodes the messages with accelerometers. Ripple thus requires physical contact with accelerometers to sense vibrations and decode messages. Ripple II [37] builds a faster vibration communication channel by re-designing and optimizing the OFDM-based PHY layer and the proactive MAC layer. VibSense [25] senses vibrations using touch screens. Similarly, capacitive touchscreens of smart devices are exploited to build communication channels between objects (*e.g.*, rings) and touchscreens for user authentication [42]. Skin-MIMO [31] presents a MIMO vibration communication over the skin using motors and piezo transducers. Unlike these works where vibrations propagate through solid surfaces, *mmRipple* aims to wirelessly sense smartphone vibrations using COTS mmWave radars without any physical contact.

10 CONCLUSION

In this paper, we address a series of technical challenges in designing and implementing *mmRipple*, which allows users to send messages to mmWave radars through smartphone vibrations. *mmRipple* can support concurrent reception of vibration signals from multiple smartphones by leveraging the diversities of smartphone vibrations in frequency, time, and location. *mmRipple* mitigates the impact of device movement, noise and interference with novel signal processing techniques. While future work is needed to further improve *mmRipple*, we believe it is an important step towards demonstrating the feasibility and practicality of building communication channels between vibrating objects (not limited to smartphones) and mmWave radars.

ACKNOWLEDGMENTS

This work is supported in part by Hong Kong General Research Fund (GRF) under grant PolyU 152165/19E, and National Natural Science Foundation of China under grant U21A20462, 62176205. Yuanqing Zheng and Jinsong Han are the corresponding authors.

REFERENCES

[1] [n. d.]. TI mmWave Antennas. Instruments, Texas. https://dev.ti.com/tirex/explore/node?node=A__AB9Ek2o7Z7KxVGvvfm7HCA__radar_toolbox__1AslXXD__LATEST
[2] [n. d.]. YV400 Portable Vibrometer. https://www.amazon.com/Portable-Vibrometer-Vibration-Integrated-Thermal/dp/B092VLT33R
[3] 2022. Vibrator. In *Android Developers*. https://developer.android.com/reference/android/os/Vibrator
[4] Kang Min Bae, Namjo Ahn, Yoon Chae, Parth Pathak, Sung-Min Sohn, and Song Min Kim. 2022. OmniScatter: extreme sensitivity mmWave backscattering using commodity FMCW radar. In *Proceedings of MobiSys*.
[5] Atul Bansal, Akshay Gadre, Vaibhav Singh, Anthony Rowe, Bob Iannucci, and Swarun Kumar. 2021. Owll: Accurate lora localization using the tv whitespaces. In *Proceedings of ACM IPSN*.
[6] Xu Chen, Zhiyong Feng, Zhiqing Wei, Ping Zhang, and Xin Yuan. 2021. Code-division OFDM joint communication and sensing system for 6G machine-type communication. *IEEE Internet of Things Journal* (2021).
[7] Jungchan Cho, Inhwan Hwang, and Songhwai Oh. 2012. Vibration-based surface recognition for smartphones. In *Proceedings of IEEE RTCSA*.
[8] Wang Dan. 2018. Cascaded Radar And Body and Chassis Automotive Applications. *Texas Instruments (TI) mmWave Training Series* (2018).
[9] Han Ding, Lei Guo, Cui Zhao, Fei Wang, Ge Wang, Zhiping Jiang, Wei Xi, and Jizhong Zhao. 2020. RFnet: Automatic gesture recognition and human identification using time series RFID signals. *Mobile Networks and Applications* 25 (2020), 2240–2253.
[10] Jian Gong, Xinyu Zhang, Kaixin Lin, Ju Ren, Yaoxue Zhang, and Wenxun Qiu. 2021. RF vital sign sensing under free body movement. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–22.
[11] Jeremy Gummeson, James Mccann, Chouchang Yang, Damith Ranasinghe, Scott Hudson, and Alanson Sample. 2017. RFID light bulb: Enabling ubiquitous deployment of interactive RFID systems. *Proceedings of ACM Interact. Mob. Wearable Ubiquitous Technol.* (2017).
[12] Junchen Guo, Meng Jin, Yuan He, Weiguo Wang, and Yunhao Liu. 2021. Dancing Waltz with Ghosts: Measuring Sub-mm-Level 2D Rotor Orbit with a Single mmWave Radar. In *Proceedings of ACM IPSN*.
[13] Unsoo Ha, Salah Assana, and Fadel Adib. 2020. Contactless seismocardiography via deep learning radars. In *Proceedings of ACM MobiCom*.
[14] Aboulnasr Hassanien, Moeness G Amin, Yimin D Zhang, and Fauzia Ahmad. 2015. Dual-function radar-communications: Information embedding using sidelobe control and waveform diversity. *IEEE Transactions on Signal Processing* (2015).
[15] Kathy J Horadam. 2012. Hadamard matrices and their applications. In *Hadamard Matrices and Their Applications*.
[16] Ningning Hou, Xianjin Xia, Yifeng Wang, and Yuanqing Zheng. 2023. One Shot for All: Quick and Accutate Data Aggregation for LPWANs. In *Proceedings of IEEE INFOCOM*.
[17] Sungjae Hwang and Kwangyun Wohn. 2013. Vibrotactor: Low-cost placement-aware technique using vibration echoes on mobile devices. In *Proceedings of ACM IUI*.
[18] Cesar Iovescu and Sandeep Rao. 2017. The fundamentals of millimeter wave sensors. *Texas Instruments* (2017).
[19] Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. 2020. mmVib: micrometer-level vibration measurement with mmwave radar. In *Proceedings of ACM MobiCom*.
[20] Haotian Jiang, Jiacheng Zhang, Xiuzhen Guo, and Yuan He. 2021. Sense Me on the Ride: Accurate Mobile Sensing over a LoRa Backscatter Channel. In *Proceedings of ACM SenSys*.
[21] Abdelwahed Khamis, Branislav Kusy, Chun Tung Chou, Mary-Louise McLaws, and Wen Hu. 2020. RFWash: a weakly supervised tracking of hand hygiene technique. In *Proceedings of ACM SenSys*.
[22] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.
[23] Hyewon Lee, Tae Hyun Kim, Jun Won Choi, and Sunghyun Choi. 2015. Chirp signal-based aerial acoustic communication for smart devices. In *Proceedings of IEEE INFOCOM*.
[24] Fan Liu, Longfei Zhou, Christos Masouros, Ang Li, Wu Luo, and Athina Petropulu. 2018. Toward dual-functional radar-communication systems: Optimal waveform design. *IEEE Transactions on Signal Processing* (2018).
[25] Jian Liu, Yingying Chen, Marco Gruteser, and Yan Wang. 2017. VibSense: Sensing Touches on Ubiquitous Surfaces through Vibration. In *Proceedings of IEEE SECON*.
[26] Jun Liu, Jiayao Gao, Sanjay Jha, and Wen Hu. 2021. Seirios: leveraging multiple channels for LoRaWAN indoor and outdoor localization. In *Proceedings of ACM MobiCom*.
[27] Jianwei Liu, Yinghui He, Chaowei Xiao, Jinsong Han, Le Cheng, and Kui Ren. 2022. Physical-World Attack towards WiFi-based Behavior Recognition. In *Proceedings of IEEE INFOCOM*.

[28] Tiantian Liu, Ming Gao, Feng Lin, Chao Wang, Zhongjie Ba, Jinsong Han, Wenyao Xu, and Kui Ren. 2021. Wavoice: A Noise-resistant Multi-modal Speech Recognition System Fusing mmWave and Audio Signals. In *Proceedings of ACM SenSys*.

[29] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A Stankovic, Niki Trigoni, and Andrew Markham. 2020. See through smoke: robust indoor mapping with low-cost mmwave radar. In *Proceedings of ACM MobiSys*.

[30] Chris Xiaoxuan Lu, Muhamad Risqi U Saputra, Peijun Zhao, Yasin Almalioglu, Pedro PB de Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, and Andrew Markham. 2020. milliEgo: single-chip mmWave radar aided egomotion estimation via deep sensor fusion. In *Proceedings of ACM SenSys*.

[31] Dong Ma, Yuezhong Wu, Ming Ding, Mahbub Hassan, and Wen Hu. 2020. Skin-MIMO: Vibration-based MIMO Communication over Human Skin. In *Proceedings of IEEE INFOCOM*.

[32] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. 2018. SignFi: Sign language recognition using WiFi. *Proceedings of ACM Interact. Mob. Wearable Ubiquitous Technol.* (2018).

[33] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Ramarathnam Venkatesan. 2013. Dhwani: Secure Peer-to-Peer Acoustic NFC. *Proceedings of ACM SIGCOMM* (2013).

[34] John Nolan, Kun Qian, and Xinyu Zhang. 2021. RoS: passive smart surface for roadside-to-vehicle communication. In *Proceedings of ACM SIGCOMM*.

[35] Jacopo Pegoraro, Jesus O Lacruz, Michele Rossi, and Joerg Widmer. 2022. SPARCS: A Sparse Recovery Approach for Integrated Communication and Human Sensing in mmWave Systems. In *Proceedings of ACM IPSN*.

[36] Moniara R Romero, Ram M Narayanan, Erik H Lenzing, Daniel C Brown, and Kristoffer L Greenert. 2020. Wireless underwater-to-air communications via water surface modulation and radar detection. In *Radar Sensor Technology XXIV*, Vol. 11408. SPIE, 58–71.

[37] Nirupam Roy and Romit Roy Choudhury. 2016. Ripple II: Faster Communication through Physical Vibration. In *Proceedings of USENIX NSDI*.

[38] Nirupam Roy, Mahanth Gowda, and Romit Roy Choudhury. 2015. Ripple: Communicating through Physical Vibration. In *Proceedings of USENIX NSDI*.

[39] Xian Shuai, Yulin Shen, Yi Tang, Shuyao Shi, Luping Ji, and Guoliang Xing. 2021. millieye: A lightweight mmwave radar and camera fusion system for robust object detection. In *Proceedings of ACM/IEEE IoTDI*.

[40] Elahe Soltanaghaei, Akarsh Prabhakara, Artur Balanuta, Matthew Anderson, Jan M Rabaey, Swarun Kumar, and Anthony Rowe. 2021. Millimetro: mmWave retro-reflective tags for accurate, long range localization. In *Proceedings of ACM MobiCom*. 69–82.

[41] Francesco Tonolini and Fadel Adib. 2018. Networking across boundaries: enabling wireless communication through the water-air interface. In *Proceedings of ACM SIGCOMM*.

[42] Tam Vu, Akash Baid, Simon Gao, Marco Gruteser, Richard Howard, Janne Lindqvist, Predrag Spasojevic, and Jeffrey Walling. 2012. Distinguishing Users with Capacitive Touch Communication. In *Proceedings of ACM MobiCom*.

[43] Chao Wang, Feng Lin, Tiantian Liu, Kaidi Zheng, Zhibo Wang, Zhengxiong Li, Ming-Chun Huang, Wenyao Xu, and Kui Ren. 2022. mmEve: Eavesdropping on Smartphone's Earpiece via COTS mmWave Device. In *Proceedings of ACM MobiCom*.

[44] Fei Wang, Sanping Zhou, Stanislav Panev, Jinsong Han, and Dong Huang. 2019. Person-in-WiFi: Fine-grained person perception using WiFi. In *Proceedings of IEEE/CVF ICCV*.

[45] Ge Wang, Chen Qian, Kaiyan Cui, Xiaofeng Shi, Han Ding, Wei Xi, Jizhong Zhao, and Jinsong Han. 2020. A universal method to combat multipaths for rfid sensing. In *Proceedings of IEEE INFOCOM*.

[46] Weiguo Wang, Jinming Li, Yuan He, Xiuzhen Guo, and Yunhao Liu. 2022. MotorBeat: Acoustic Communication for Home Appliances via Variable Pulse Width Modulation. *Proceedings of ACM Interact. Mob. Wearable Ubiquitous Technol.* (2022).

[47] Xuanzhi Wang, Kai Niu, Jie Xiong, Bochong Qian, Zhiyun Yao, Tairong Lou, and Daqing Zhang. 2022. Placement Matters: Understanding the Effects of Device Placement for WiFi Sensing. *Proceedings of ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 1 (2022), 1–25.

[48] Yanwen Wang, Jiannong Cao, and Yuanqing Zheng. 2021. Toward a Low-Cost Software-Defined UHF RFID System for Distributed Parallel Sensing. *IEEE Internet of Things Journal* 8, 17 (2021), 13664–13676. https://doi.org/10.1109/JIOT.2021.3067379

[49] Greg Welch, Gary Bishop, et al. 1995. An introduction to the Kalman filter. (1995).

[50] Chenhan Xu, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Huining Li, Chen Song, Kun Wang, and Wenyao Xu. 2019. Waveear: Exploring a mmwave-based noise-resistant speech sensing for voice-user interface. In *Proceedings of ACM MobiSys*.

[51] Hongfei Xue, Yan Ju, Chenglin Miao, Yijiang Wang, Shiyang Wang, Aidong Zhang, and Lu Su. 2021. mmMesh: towards 3D real-time dynamic human mesh construction using millimeter-wave. In *Proceedings of ACM MobiSys*.

[52] Qiang Yang, Kaiyan Cui, and Yuanqing Zheng. 2023. VoShield: Voice Liveness Detection with Sound Field Dynamics. In *Proceedings of IEEE INFOCOM*.

[53] Qiang Yang and Yuanqing Zheng. 2022. Deepear: Sound localization with binaural microphones. *IEEE Transactions on Mobile Computing* (2022).

[54] Xin Yang, Jian Liu, Yingying Chen, Xiaonan Guo, and Yucheng Xie. 2020. MU-ID: Multi-user identification through gaits using millimeter wave radios. In *Proceedings of IEEE INFOCOM*.

[55] Yanni Yang, Huafeng Xu, Qianyi Chen, Jiannong Cao, and Yanwen Wang. 2023. Multi-Vib: Precise Multi-point Vibration Monitoring Using mmWave Radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023), 1–26.

[56] Zhicheng Yang, Parth H Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. 2017. Vital sign and sleep monitoring using millimeter wave. *ACM Transactions on Sensor Networks* (2017).

[57] Fusang Zhang, Zhaoxin Chang, Kai Niu, Jie Xiong, Beihong Jin, Qin Lv, and Daqing Zhang. 2020. Exploring LoRa for Long-range Through-wall Sensing. *Proceedings of ACM Interact. Mob. Wearable Ubiquitous Technol.* 4 (2020), 1 – 27.

[58] Jin Zhang, Zhuangzhuang Chen, Chengwen Luo, Bo Wei, Salil S Kanhere, and Jianqiang Li. 2022. MetaGanFi: Cross-Domain Unseen Individual Identification Using WiFi Signals. *Proceedings of ACM Interact. Mob. Wearable Ubiquitous Technol.* (2022).

[59] Qixun Zhang, Hongzhuo Sun, Xinye Gao, Xinna Wang, and Zhiyong Feng. 2022. Time-division ISAC Enabled Connected Automated Vehicles Cooperation Algorithm Design and Performance Evaluation. *IEEE Journal on Selected Areas in Communications* (2022).

[60] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. 2019. mid: Tracking and identifying people with millimeter wave radar. In *Proceedings of IEEE DCOSS*.