TagController: A Universal Wireless and Battery-free Remote Controller using Passive RFID Tags

Dong Li[†], Feng Ding[†], Qian Zhang[†], Run Zhao[‡], Jinshi Zhang[†] and Dong Wang[†*]

[†]School of Software, Shanghai Jiao Tong University, China
[‡]Computer Science Department, Shanghai Jiao Tong University, China
{dong.l,dingfengstju,qwert3472,zhangjinshi,wangdong}@sjtu.edu.cn
zhaorun@cs.sjtu.edu.cn

ABSTRACT

Innovative Human Machine Interface technologies are fundamentally reshaping the way people live, entertain and work. Passive RFID tags, benefiting from its wireless, inexpensive and battery-free sensing ability, are gradually being applied in new-style interaction interfaces, ranging from virtual touch screen to 3D mouse. This paper presents TagController, a universal wireless and battery-free remote controller with two types of interactive actions. The key insight is that the fine-grained phase information extracted from RF signals is capable of perceiving various actions. TagController can recognize 10 actions without any training or prestored profiles by executing a sequence of functional components, *i.e.* preprocessor, action detector and action recognizer. We have implemented TagController with COTS RFID devices and conducted substantial experiments in different scenarios. The results demonstrate that TagController can achieve an average recognition accuracy of 95.8% and 94.3% in the scenarios of one and two remote controllers, respectively, which promises its feasibility and robustness.

CCS CONCEPTS

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

KEYWORDS

Battery-free remote controller, action recognition, COTS RFID

ACM Reference format:
Dong Li[†], Feng Ding[†], Qian Zhang[†], Run Zhao[‡], Jinshi Zhang[†] and Dong Wang[†]. 2017. TagController: A Universal Wireless and Battery-free Remote Controller using Passive RFID Tags. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Melbourne, VIC, Australia, November 7–10, 2017 (MobiQuitous 2017),* 10 pages.
https://doi.org/10.1145/3144457.3144498

*Dong Wang is the corresponding author.

MobiQuitous 2017, November 7–10, 2017, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5368-7/17/11…$15.00
https://doi.org/10.1145/3144457.3144498

1 INTRODUCTION

Incessant innovations in areas, such as wireless sensing, gesture recognition and wearable devices, are incorporating with even newer technologies like machine learning to open up endless possibilities for novel interaction interfaces.

Existing unconventional interaction interfaces can be mainly classified into three categories according to their supporting technologies: *Computer vision based interfaces*, such as Kinect [18], RealSense [10], Leap Motion [11], are fused with multifarious sensing technologies to achieve human-computer interaction. Although these interfaces can create amazing user experiences, they have a strong dependence on light and may raise serious privacy concerns; *Sensor based interfaces* leverage integrated sensors embedded in smart phones (*e.g.* accelerometer and gyroscope [3], magnetometer [1] and microphone [17]) or dedicated sensors (*e.g.* SmartSkin sensor [21]) deployed in environments to interact with humans. These interfaces are either energy-consuming or labor-intensive; *Wi-Fi based interfaces* offer opportunities for new-fashioned interaction modes, *e.g.* preforming device-free gesture control [2], playing a contactless exergame [20] and striking a printed keyboard [5]. These interfaces are quite intriguing and encouraging, but they are susceptible to environments and thus have some restrictions in practical use.

Passive Radio Frequency Identification (RFID) tags, benefiting from its wireless, inexpensive and battery-free sensing ability, are gradually being applied in new-style interaction interfaces, such as 3D mouse [15], virtual touch screen [23, 26], interactive light controller [12], *etc.* Inspired by these glittering interfaces, we devise TagController, the first universal wireless and battery-free remote controller for handling ubiquitous human to machine interactions via COTS RFID devices. As is known to all, remote controllers have permeated almost every facet of our lives, including operating television, controlling slideshow, playing video games, *etc.* Our vision is that TagController can provide several attractive advantages over traditional remote controllers: 1) **Battery-free and Inexpensive:** Passive RFID tags are equipped without any batteries, which apparently eliminates the trouble for replacing batteries as well as protects the environment. Furthermore, there is no need to do any hardware modifications with COTS RFID readers and antennas. [1] Thus, the total cost for one remote controller is less than a half US dollar (4 tags with 5-10 cents each [28]), which makes it possible

[1]As assumed in GRfid [30] and Tagball [15], RFID readers are deployed as basic equipments in indoor environment.

for fabricating one remote controller for each machine; 2) **Easy-to-find and Easy-to-make:** It always happens that people restlessly look for the remote controller while it is not in the room at all. TagController is able to assist people in judging whether the remote controller is nearby rapidly. Even though the remote controller is missing, it is rather simple and economic to refabricate a new one (just attaching 4 tags on a cardboard); 3) **Light-insensitive and Multidevice-supported:** TagController inherits a preponderance of being insensitive to light from RF based systems, *i.e.* it can be utilized in a relative gloomy environment. Moreover, since actions performed by users via the self-made remote controller are carefully designed according to factors that dominate phase changes, it is much more tolerant of ever-changing surroundings than device-free interaction interfaces, and thereby multiple remote controllers can work simultaneously.

The key insight of TagController is that the reliable and fine-grained phase information extracted from RF signals is capable of perceiving various actions when users operate the battery-free remote controller. More specifically, each action can induce distinctive phase fluctuations, which can be regarded as a particular indicator for further recognition. Although the rationale behind sounds fairly intuitive, there are still some rigorous challenges needed to be addressed:

(1) *How to recognize actions without any training or prestored profiles?* There is no doubt that it is onerous to construct suitable phase profiles and also time-consuming to match real-time phase extraction with phase profiles, indicating that phase profiles are infeasible for real-time action recognition in TagController.

As shown in Figure 1, we address this problem by meticulously excogitating 10 interactive actions, including 4 press actions and 6 motions, on the basis of deliberating over the factors that dominate phase changes (§2.1). Unfortunately, these ten actions cannot be distinguished from each other with phase information from one tag, thus we propose action-specific features by integrating phase changes of four tags to exclusively discern different actions (§3.3).

(2) *How to exclude the influence of disparate phase patterns engendered by individual diversity?* Our extensive experiments have manifested that users with various habits, such as the duration of one action and movement range, can give rise to totally different phase patterns when they perform the same action. Actually, even if a user performs the same action several times, it is difficult to guarantee that the phase patterns are identical.

After doing substantial analysis of the experiments, we present square-array tag placement for the sake of generating distinct phase profiles for each action (§2.2). Most importantly, action-specific features constructed based on this special tag placement are completely immune from the problem mentioned above (§3.3).

(3) *How to make sure that actions can be recognized fast and precisely?* As an instant interaction interface, it is of great importance for TagController to provide real-time and precise feedbacks.

We implement TagController as a pipeline containing three components: preprocessor (§3.1), FSM based action detector (§3.2) and action recognizer (§3.3). In preprocessor, processed tag readings are compressed frame by frame to reduce computational overhead. In action detector, the frame Mean Absolute Deviation (MAD) fed by preprocessor can be regarded as an instant input for a simple Finite State Machine which can detect an action almost in real-time. In action recognizer, benefiting from action-specific features, actions can be precisely recognized without matching prestored profiles, which definitely speeds up the process of recognition.

In summary, our main contributions can be concluded as follows:

- To the best of our knowledge, TagController is the first universal wireless and battery-free remote controller for handling ubiquitous human-to-machine interactions. This system possesses multitudinous target applications and has revealed its tempting superiority, such as being easy to find and make, supporting multidevice.
- We sufficiently investigate factors that dominate phase changes and accordingly devise a series of actions which can be recognized without phase profiles. In addition, we propose action-specific features extracted from all the tags on the remote controller to uniquely discriminate actions with the help of special square-array tag placement.
- We implement TagController with COTS RFID devices and self-made remote controllers. Experimental results show that TagController can achieve an average recognition accuracy of 95.8% and 94.3% in the scenarios of one and two remote controllers, respectively.

The rest of this paper is organized as follows. Section 2 presents the overview of the system. Detailed designs of each processing module are described in Section 3. Section 4 and Section 5 demonstrate the implementation and evaluation of our system respectively. Section 6 reviews the related work our system draws on. Finally, we conclude our work in Section 7.

2 SYSTEM OVERVIEW

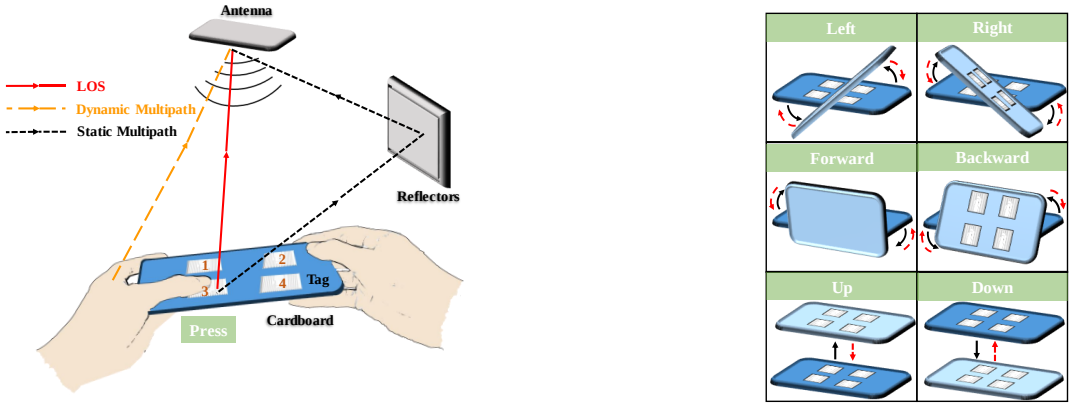
In this section, we first present the communication model of our passive Ultra High Frequency (UHF) RFID based system and elaborate on the design of non-training actions. Next, we perform two sets of preliminary experiments using a COTS ImpinJ reader and several commercial tags to expound the reason why we should bring in the group sensing technique. Finally, we roughly introduce the overall work-flow of our system.

2.1 Communication Model and Action Design

A passive UHF RFID tag interacts with a reader via backscatter radio links. The tag with no battery equipped, rectifies the received power emitted by the reader to support operation of its circuitry, and modulates the transmitted signal from the reader in order to send information back [7]. There are two magic weapons for present COTS UHF readers to be qualified for ubiquitous computing: 1) *Long-range read and write:* they can interrogate tags within a few dozen meters; 2) *Detailed low level data:* they have the ability to proffer ample low level user data including RF phase, RSSI and Doppler shifts [9]. In our system, the reliable and fine-grained phase information is employed as the only perceptive indicator in virtue of its sensitivity to movement and multipath [19].

The integrated signal $S(t)$ received by the reader can be represented by the following equation:

$$S(t) = A(t)e^{-j\varphi(t)} \tag{1}$$



(a) Communication model. (b) Motions.

Figure 1: Communication model and Actions. There are 10 actions in total, including 4 presses corresponding to the number of tags (shown in left) and 6 motions corresponding to three dimensions (shown in right). Each action consists of two steps: 1) Presses: put the finger on the tag and then pull the finger up; 2) Motions: perform the step displayed by the black solid line and then perform the step by the red dotted line.

where $A(t)$ is the complex valued representation of attenuation and the initial phase offset and $\varphi(t)$ denotes the time-varying phase offset.

As shown in Figure 1(a), the integrated signal $S(t)$ is comprised of the ones that travel along the direct path between the antenna and the tag, and those that are reflected, *i.e.* multipath. The multipath can be divided into two parts: the ones reflected by the ubiquitous and static reflectors like floors and walls (static multipath) and those reflected by moving objects like hand and body (dynamic multipath). Therefore, the integrated signal $S(t)$ can also be expressed as follows:

$$S(t) = S_{\text{dir}}(t) + S_{\text{sta}}(t) + S_{\text{dyn}}(t) \tag{2}$$

where $S_{\text{dir}}(t)$, $S_{\text{sta}}(t)$ and $S_{\text{dyn}}(t)$ stand for the signal of direct path, the signals reflected by static reflectors and dynamic ones respectively.

The phase rotation of the direct path $\varphi_{\text{dir}}(t)$ is associated with the distance between the reader and the tag $d(t)$ as well as some hardware characteristics c [9], which can be further denoted as

$$\varphi_{\text{dir}}(t) = \frac{4\pi}{\lambda} d(t) + c \tag{3}$$

When putting all the three equation above together, the following equation can be inferred:

$$A(t)e^{-j\varphi(t)} = A_{\text{dir}}(t)e^{-j\frac{4\pi}{\lambda}d(t)+c} + S_{\text{sta}}(t) + S_{\text{dyn}}(t) \tag{4}$$

where $A_{\text{dir}}(t)$ denotes the complex valued representation of attenuation in the direct path.

Enlightened by Equation 4, we ingeniously devise two types of non-training actions whose phase-changing patterns can be easily identified: 1) ***Press:*** When a finger approaches a tag and then moves away from it as shown in Figure 1(a), the phase of this pressed tag changes observably as shown in Figure 2(b), which is primarily caused by the variation of $S_{\text{dyn}}(t)$. It is noteworthy that there is missing tag read during the period of pressing, which will be explained in §3.1. 2) ***Motion:*** When a user performs any action depicted in Figure 1(b), the phases of all tags change regularly along with the time-varying distance $d(t)$ as shown in Figure 2(b).

These actions provide two prominent benefits: On one hand, they can raise the ability of being insensitive to ever-changing surroundings, which is accomplished by directly manipulating tags; On the other hand, phase patterns of each action are distinct enough to be separated without training any phase profiles.

2.2 Preliminary Experiments

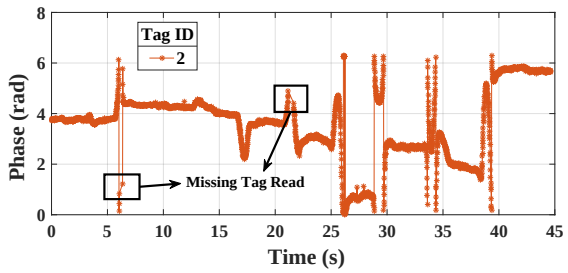
In order to decipher the reason why we should introduce the group sensing technique, we conduct two sets of preliminary experiments using COTS RFID devices. These two experimental settings are as follows: As shown in Figure 2(a), a self-made remote controller are attached with 4 tags and an antenna connected to an ImpinJ reader is fixed above the remote controller with a distance of 1.5m.

Observation 1. *It is insufficient to recognize 10 actions merely leveraging the phase information from one tag.*

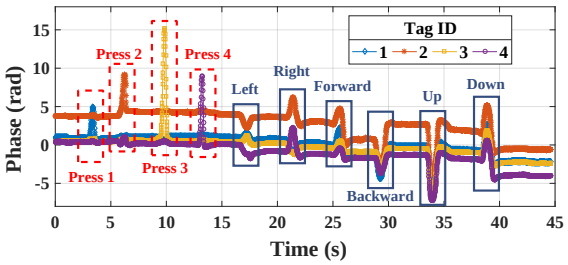
We ask a volunteer to perform 10 exemplary actions one by one, including 4 presses and 6 motions. As shown in Figure 2(a), phase changes of one tag engendered by these actions are quite clear and obvious. However, these actions cannot be distinguished from each other on account of their non-unique phase patterns. In order to solve this problem, we arrange four tags at the vertex of a square for the sake of generating distinct phase patterns for each action when taking phase changes of all tags into consideration as depicted in Figure 2(b).

Observation 2. *One action performed by various users may be totally different due to their personal habits like range of movement and rotation axis. Actually, it is hard to guarantee that phase patterns are identical even when a user repeatedly perform the same action.*

Five volunteers are invited to perform each of ten actions twice in turn. Three representative phase profiles of action 'Left' are selected to illustrate the observation mentioned above. Except that in rightmost figure, the remaining phase profiles in Figure 3 are produced by the same user. In order to solve this problem, we design



(a) One of raw phase profiles.



(b) Processed phase profiles.

Figure 2: The phase profiles of four tags are collected after a user performs ten actions.

the group sensing technique by leveraging several action-specific features to exclude disparate phase patters of the same action (§3.3).

2.3 System Flow

Figure 4 demonstrates the overall framework of TagController, containing three parts, *i.e.* hardware components, softwares and user-defined applications.

Hardware: Almost all the hardwares adopted in TagController (*e.g.* multiple tags, a RFID reader, an antenna and one PC) are off-the-shelf except for the self-made remote controller. As depicted in Figure 4, it is effortless to fabricate a battery-free remote controller, just attached with four square-array tags on the cardboard.

Software: Upon receiving data packets from the reader, the PC feeds them into three data processors in sequence, that is, preprocessor, action detector and action recognizer. The *preprocessor* unwraps the raw phase readings, fills the missing ones and most importantly compresses the processed phase readings frame by frame. Next, the *action detector* monitors the potential actions by executing a Finite State Machine using a special indicator from the preprocessor. Then it checks the validity of possible actions and passes compressed action profiles to the next component if actions are valid. Finally, the *action recognizer* extracts action-specific features from action profiles and then classifies the corresponding action by performing a simple action recognition algorithm.

User-defined applications: In order to demonstrate the feasibility and availability of TagController, we developed a simple application which can display real-time actions on the screen. It is indisputable that developers can rapidly build self-defined applications based on TagController.

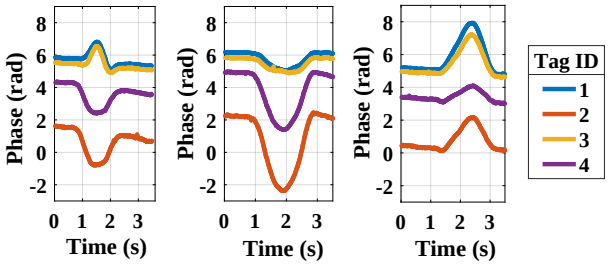


Figure 3: Left action: The first two phase profiles comes from the same user while the remaining one from the other.

3 METHODOLOGY

This section details the design of TagController, including problems arising when we try to design each component, key observations obtained by exhaustively analyzing these problems and core techniques elaborately designed according to corresponding observations.

3.1 Preprocessor

The preprocessor is an indispensable component in our system for the inherent slight imperfections of RFID systems: 1) Raw phase readings is likely to be wrapped due to the hardware characteristics of commercial readers; 2) There are two kinds of cases where phase readings will be missing for some time; 3) The millisecond-level interrogation between two tag reads specified by the C1G2 protocol [8] makes it hard to meet the requirements of real-time operation if each module processes the data one by one. All of these three foibles will make the system fail to work practically. Therefore, we remedy the limitations by orderly executing three modules, including phase unwrapping, phase filling as well as frame compression.

Phase unwrapping: As shown in Figure 2(a), the phase readings returned by ImpinJ readers is a periodic function with period 2π radians [9], which is definitely inconvenient for subsequent analysis. Thus, the first priority of preprocessor is to unwrap the raw phase readings. As assumed in TagBooth [16], the absolute difference of two adjacent reading phase value should be smaller than π because of the low frequency of hand movement compared with the reader interrogation. For simplicity, we adopt the approach in TagBooth [16] to solve this problem.

Phase filling: As we can observe from Figure 2(a), there are two missing tag reads in the process of performing ten actions, which exactly correspond to two different cases. The first case always happens when dielectric material like fingers comes very close to a tag [25]. And the second case is incurred by the blind direction of an RFID tag which is related with its non-isotropic radiation [27].

In order to successfully recognize actions, intact phase readings must be provided to the subsequent module namely frame compression, which can be achieved by interpolating phase values among missing readings. Unfortunately, although there exists some evidence implying missing readings, common reader interfaces do not expose this information [25]. Hence, we attempt to acquire the maximum interval between two tag reads through experiments, which can help us identify missing readings. Extensive experiments

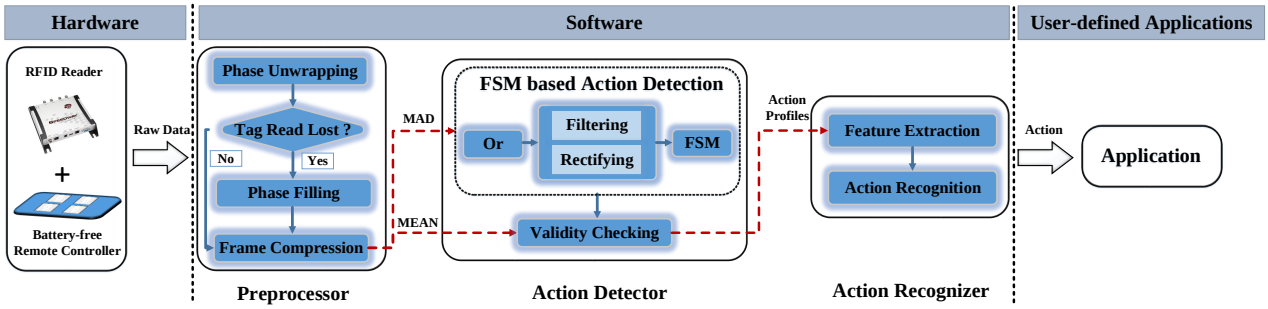
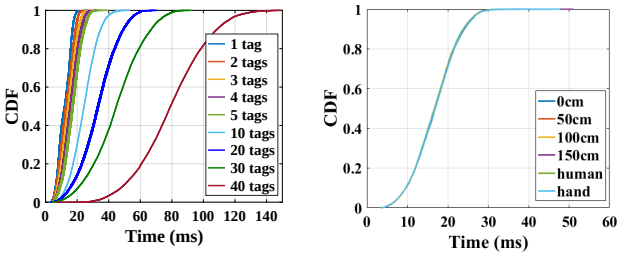


Figure 4: System architecture of TagController.



(a) The number of tags.

(b) Various distances and dynamic multipath with 4 tags.

Figure 5: Factors affecting the interval between two reads.

have been conducted with the number of tags ranging from 1 to 40, various dynamic multipath brought by humans as well as hands, and the distance between tags and the antenna ranging from 0 to $1m$. The experimental results shown in Figure 5 can be summarized as follows: 1) Compared with Figure 5(a) and Figure 5(b), we can draw a conclusion that the interval between two tag reads is almost impervious to dynamic multipath as well as the distance between tags and the antenna; 2) As we can obtain from Figure 5(b), although the maximum interval between two tag reads are positively associated with the number of tags in the interrogation zone, the relationship between them is rather stable, indicating that developers can prestore a key-value table which records the corresponding relationship.

Each time when a phase reading arrives, the phase filling module checks the interval between the current phase readings and the previous one. If the interval exceeds the maximum interval ascertaining by looking up the prestored table, a fast, efficient and stable interpolation approach called cubic spline interpolation [4] is executed by the current phase reading along with two preceding readings.

Frame Compression: The frame compression module is introduced to serve two significant purposes. *Reducing computational overhead:* The C1G2 protocol implemented by readers is capable of performing millisecond-level interrogations, suggesting that it is hard to process the phase readings one by one in each processing module. Therefore, phase readings are compressed frame by frame to meet the requirements of real-time operation as shown in Figure 6. In fact, the simple rationale behind is that the mean of phase values substitutes for all the phase values in a frame, which can not only reduce the computational cost, but also smooth the phase values. *Providing action detector with the vital indicator:* We can draw a conclusion from the basic observation that the extent of variations during actions is much more severe than that of "non-action" periods. Therefore, this module calculates the mean absolute deviation (MAD) of phase values for each frame and regards it as a vital indicator for action. For the convenience of subsequent processing, we convert the MAD of each frame into a binary value by defining a threshold mad_thresh. This threshold can be determined by preliminary measurements, which is empirically set to 0.05 in our implementation. If the MAD of frame i is greater than mad_thresh, this module replaces the original MAD with 1 and 0 otherwise. Mathematically, this process can be denoted as

$$frame\_mad_i = \begin{cases} 1, & frame\_mad_i \geqslant mad\_thresh \\ 0, & otherwise \end{cases} \quad (5)$$

where $frame\_mad_i$ represents the MAD of its corresponding frame i. It is worth noting that the frame size is a critical parameter affecting the performance of our system, so we test various frame sizes and empirically set it as $100ms$ which optimizes the performance. Figure 6 shows the result of performing frame compression, which verifies its effectiveness.

3.2 Action Detector

After preprocessing the raw phase readings, the action detector is applied to probe potential actions. It is fairly essential yet challenging when taking the accuracy and instantaneity into account. As a matter of fact, there are many proven segmentation techniques such as KL divergence [6], foreground detection [24], Modified Varri Method [30] and so on. In this paper, we devise a FSM based action detection algorithm exploiting the distinction of signal features. Furthermore, to increase the reliability of our algorithm, we make a second examination by verifying the validity of possible actions.

FSM based action detection: Owing to the discrepancy of movement between tags when performing actions, their MADs for the same frame may be divergent, thus we perform a logical or of all the MADs in the same frame as shown in Figure 6. However, there are still two problems needed to be addressed albeit the resulting MAD reveals itself as a useful indicator for actions. On one hand, some false positive outliers may appear when users casually shake their hands or bodies, which must be filtered before being leveraged; On the other hand, the peculiar moving patterns

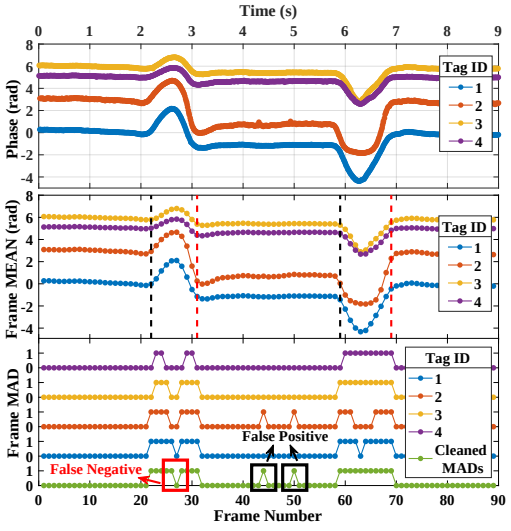


Figure 6: Snapshot of frame MAD, frame Mean and segmentation results when performing two actions.

Algorithm 1 Algorithm for filtering and rectifying false MADs

Input: Three adjacent frame MADs $frame\_mad_{i-1}$, $frame\_mad_i$ and $frame\_mad_{i+1}$
Output: The processed frame MAD $frame\_mad_i$

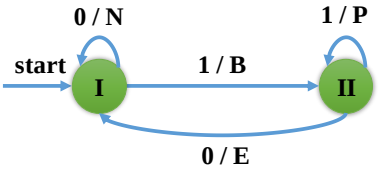
```
frame_mad_sum = frame_mad_{i-1} + frame_mad_{i+1}
if frame_mad_i == 1 then
    // Filtering false positive MADs
    if frame_mad_sum == 0 then
        frame_mad_i = 0
    end if
else
    // Rectifying false negative MADs
    if frame_mad_sum == 2 then
        frame_mad_i = 1
    end if
end if
```

of actions give rise to some false negative MADs, which may take place in the middle of actions and should be undoubtedly rectified to insure that actions are segmented as a whole. After filtering or rectifying a faulty MADs, we send this cleaned MAD to a simple Finite State Machine for action detection.

Logical or operation: Once MADs from all tags in the same frame have been collected, the action detector calculates the result of the logical or operation and delivers it to the next module.

Filtering and Rectifying: In order to avoid detecting false actions or splitting an unbroken action into two parts, the action detector filters the false positive MAD and rectifies the false negative MAD through the approach described in Algorithm 1.

Finite State Machine: After collecting the cleaned MAD, a simple Finite State Machine is utilized to detect an action. As shown in Figure 7, when the FSM is in the static state denoted as I, if the cleaned MAD equals to one, the FSM informs the validity checking



State	
I	Static state
II	Kinetic state

Output	
N	No action
B	Beginning an action
P	Performing an action
E	Ending an action

Figure 7: A simple Finite State Machine.

module to store the mean of the current frame; otherwise, the validity checking module should discard it. When the FSM is in the kinetic state denoted as II, if the cleaned MAD is equivalent to zero, the validity checking module will receive an action-detected signal; otherwise, it keeps storing the current value.

Validity Checking: Although most of outliers have been eliminated by the filtering operation, there are a few cases where users persistently shaking their bodies, which cannot be effectively handled. Thus, the action detector excludes false actions by examining the range and duration of the possible action. Finally, if this possible action passes the examination, extracted phase profiles of the action will be sent to the action recognizer. It is necessary to account for the structure of action profiles, which is a $n \times m$ matrix consisting of frame MEANs for n tags during this action. Each row of this matrix contains m frame MEANs for a specific tag.

3.3 Action Recognizer

As the core component for TagController, the action recognizer fulfills two essential functions by devising several action-specific features, including eliminating the discrepancy of phase patterns engendered by individual diversity and recognizing all the defined actions without any training or prestored profiles.

Feature extraction: Different from common activity recognition systems which need either to train the model [29] or to prestore the signal profiles [22], we come up with an audacious idea for TagController: Why not design a system with no need for extensive preliminary work? Fortunately, benefiting from meticulous designs of actions and tag placement, there are indeed some action-specific features which exclusively belong to the corresponding actions. The key insight behind is that TagController maximizes the characteristic difference of actions through treating four tags on the cardboard as a whole instead of extracting their features independently.

For the convenience of calculating the action-specific features, the Direct Current (DC) component of each tag in action profiles should be removed by subtracting the first value of each row from all the frame MEANs in this row respectively.

Feature 1 - Peak-to-Peak Amplitude (P2PA): As shown in Figure 2(b), the peak-to-peak amplitude of each tag in action profiles computed by Equation 6 is an important indicator for precise recognition.

$$P2PA = max(frame\_mean) - min(frame\_mean) \quad (6)$$

where $frame\_mean$ represents frame MEANs for each tag corresponding to each row in action profiles which has been particularly explained in section 3.2. $max()$ and $min()$ calculate the maximum and the minimum of $frame\_mean$ respectively.

Feature 2 - P2PA Divergence (PD): Thanks to the ingenious tag placement and well-designed actions, the values of *P2PA* for all tags in a specific action comply with a distinct and certain phase pattern. More specifically, in a press action, it is quite intuitive that the value of *P2PA* for the pressed tag must be far larger than the rest of tags whose values of *P2PA* are extremely small; in other actions, there is no such large division among the values of *P2PA* because at least two tags are behaving in the same way. Therefore, the press action can be differentiated from other actions by calculating the value of *PD* as follows:

$$PD = \sum_{i=1}^{n} \frac{max(P2PAs)}{P2PAs_i} \tag{7}$$

where *P2PAs* denotes a set of *P2PA* values. Accordingly, $P2PAs_i$ is the value of *P2PA* for tag i and n represents the number of tags on the remote controller. It is straightforward to determine the pressed tag immediately after detecting a press action.

Feature 3 - P2PA Product (PP): We can image that both of the up action and the down action can evoke larger phase fluctuations than other motions, namely 'Left', 'Right', 'Forward' and 'Backward'. Moreover, all the tags of these two actions behave in almost the same way, and thereby the product of all the elements in *P2PAs* will be much greater than others.

$$PP = \prod_{i=1}^{n} P2PAs_i \tag{8}$$

Feature 4 - Frame Mean Magnitude (FMM): The magnitude of $frame\_mean$ for each tag defined in Equation 9 is very useful to distinguish the up action and the down action. After calculating each row of *FMM* in actions profiles, we store them in a set called *FMMs*.

$$FMM = sum(frame\_mean) \tag{9}$$

where $sum()$ computes the sum of elements in $frame\_mean$.

Feature 5 - FMM Difference (FMMD): We leverage the remote controller depicted in Figure 1(a) to clearly explain how tag placement is used to devise the most artful feature. *FMMD* contains the *FFM* difference for combinations of tag 1 and tag 3, tag 2 and tag 4, 1 and tag 2 as well as 3 and tag 4 in sequence. When a left or right action is performed, one of the first two elements in *FMMD*, namely $FMMD_1$ or $FMMD_2$ must be the smallest one in *FMMD* on account of nearly identical behaviors for each combination of tags. Likewise, when a forward or backward action is performed, one of the last two elements in *FMMD* must be the smallest one.

Another trick point is that, although action profiles of 'Left', 'Right', 'Forward' and 'Backward' may be completely different due to the various rotation axis, $FMMD_3$ or $FMMD_4$ must be greater than 0 for 'Left' and be smaller than 0 for 'Right' which can be verified in Figure 3. Similarly, 'Forward' corresponds to $FMMD_1$ or $FMMD_2$ greater than 0 and 'Backward' otherwise.

Action recognition: As described in Algorithm 2, the process of recognizing actions based on action-specific features is quite simple due to the distinct feature of each action. It is noteworthy to emphasize that none of these features needs to be compared with a threshold except for *PD* and *PP*. Fortunately, *PD* and *PP* are so well-designed that the thresholds of these two features can be easily determined by a few experiments as we will discuss in section 5.1.

Algorithm 2 Algorithm for action recognition

Input: action profiles, pd_thresh, pp_thresh
Output: type of the inputting action

```
Removing the DC component of each row in action profiles
P2PAs ← calculate P2PA for each row in action profiles
Calculate PD of P2PAs
if PD > pd_thresh then
    Calculate the maximum of P2PAs
    Return 'Press' and the corrpesonding id of the maximum
else
    Calculate PP of P2PAs
    FMMs ← calculate FMM for each row in action profiles
    if PP > pp_thresh then
        // FMMs_i can be any one of elements in FMMs
        if FMMs_i < 0 then
            Return 'Up'
        else
            Return 'Down'
        end if
    else
        min_index ← the index of the minimum of FMMD
        if min_index equals to 1 or 2 then
            if FMMD_3 > 0 then
                Return 'Left'
            else
                Return 'Right'
            end if
        else
            if FMMD_1 > 0 then
                Return 'Forward'
            else
                Return 'Backward'
            end if
        end if
    end if
end if
```

4 SYSTEM IMPLEMENTATION

This section presents the detailed information of the hardware and software implementation.

Hardware We build TagController on some commercial off-the-shelf RFID devices and a self-made remote controller which can be fabricated without any effort. As shown in Figure 1(a), a directional antenna (Laird S9028PCR with 9 *dBi* gain) is fixed at the ceiling of a room, connected with an ImpinJ R420 reader. The reader is implemented on the basis of C1G2 [8], and can be configured with several parameters. For the sake of maximizing the performance of TagController, the reader is set to work with the highest data rate mode called "Max Throughput" as well as the maximum transmit power 32.5 *dBm*. The remote controller is made by attaching four

Action	Left	Right	Forward	Backward	Up	Down	Press 1	Press 2	Press 3	Press 4
PD	13.662	2.835	16.613	34.691	2.743	1.653	10732.09	700.480	56104.44	17702.41
PP	1.020	0.890	4.451	0.152	94.708	501.429	0.029	0.015	0.002	0.067

Figure 8: An example of *PD* and *PP* for 10 actions.

Alien Az-9629 UHF RFID Tags with a size of $25.5mm \times 25.5mm$ on a common cardboard in the shape of square.

Software In order to instantly display the outcome of our design as well as perform sufficient analysis, we develop two sets of systems. One of them is implemented in C# based on the Low Level Reader Protocol (LLRP) for collecting data from the reader and real-timely executing three processing components. The other is developed with Matlab, which can perform off-line analysis when obtaining data from the real-time system.

5 SYSTEM EVALUATION

In this section, we first account for the effectiveness of non-training action recognition and then evaluate the performance of TagController through extensive experiments in various aspects, containing the impact of tag placement, the effect of removing individual diversity and resisting multipath as well as the feasibility of multiple remote controllers.

5.1 Effectiveness of Non-training Recognition

Our intention for this design is to recognize all the defined actions without any training or prestored profiles. However, there are two action-specific numerical features, namely *PD* and *PP*, which seemingly needs to be trained for obtaining a suitable threshold. As a matter of fact, benefiting from meticulous designs of actions, these two features are distinct enough to separate corresponding actions. To account for this, we extract an example of these two features for 10 actions performed by a user as depicted in Figure 8. The values of *PD* for all press actions are several times or even more than those of other actions. Similarly, the values of *PP* for 'Up' and 'Down' have the same effect, which indicates the effectiveness of non-training recognition for those actions. In our implementation, we set the thresholds of those two features, namely *pd_thrsh* and *pp_thresh* in Algorithm 2, as 200 and 50 respectively.

5.2 Impact of Tag Placement

There is no doubt that tag placement on the cardboard expounds significant impact of action recognition. As mentioned before, we arrange four tags at the vertex of a square for the sake of generating distinct phase profiles for each action. Thus, edge length of this square defined as the distance between geometric centers of two adjacent tags is an important factor which has a great influence on the distinctiveness degree of action-specific features.

To acquire the best tag placement of battery-free remote controllers, we invited a participant to perform each of ten actions for 20 times with square edge length varying from $2.55cm$ to $8.55cm$ (taking the length of tags into account). Experimental results shown in Figure 9 demonstrate that edge length of the square has a great impact on the actions like 'Left', 'Right', 'Forward' and 'Backward'. It is reasonable that action-specific features for these actions are designed via different behaviors between 'Left' ('Right') and 'Forward' ('Backward'). The smaller of edge length is, the fewer differences are between them, resulting in low recognition rate. However, actions, such as 'Up', 'Down' and four press actions, are much less impacted by the edge length in that their features have nothing to do with the edge length. Most importantly, the performance of TagController is almost the same in the conditions when the length of edge is set to $4cm$ and $6cm$. Thus, in order to facilitate the operation, we set $4cm$ edge length as defaults in the following experiments.

5.3 Effect of Removing Individual Diversity

As discussed in section 2.2, users perform actions in a divergent way, resulting in totally different phase profiles. Action-specific features are meticulously designed for eliminating such wide variations. To explore the robustness of this system against individual diversity, we conducted a set of experiments with 10 volunteers performing each action twenty times. We can observe from the confusion matrix shown in Figure 11 that actions, *i.e.* 'Left', 'Right', 'Forward' and 'Backward', are more likely confused among each other, which can be explained by the tendency of deflecting to the backward or forward direction when performing 'Left' and 'Right'. Likewise, there exists a tendency of deflecting to the left or right direction when performing 'Forward' and 'Backward'. We can also find that both of the average accuracy of 'Press 1' and 'Press 2' are lower than those of 'Press 3' and 'Press 4', which is caused by the larger distance between the corresponding tag and the finger. The average accuracy of all actions is 95.8% which verifies the effect of removing individual diversity.

5.4 Effect of Resisting Multipath

The insensitivity to ever-changing surroundings has been briefly touched upon in section 2.1. In order to visualize the capability of resisting multipath, we perform two kinds of experiments inspecting the results of insensitivity to static multipath like tables and dynamic multipath like moving humans. In the static multipath scenario, a user was performing actions in a place where a metallic board was laid aside. In the dynamic multipath scenario, we asked two to five volunteers walking around when a user was performing actions. Each action was performed 20 times for both of these cases. As we can observe from Figure 10, the average accuracy for each action are almost the same in normal, static multipath and dynamic multipath scenarios, which demonstrates the feasibility of utilizing battery-free remote controllers in daily life.

5.5 Feasibility of Multiple Remote Controllers

The ability for resisting multipath lays a solid foundation for developing a system in which multiple remote controllers can work simultaneously. To test the feasibility of this function, one volunteer was asked to sit very close to another. It is not difficult to understand that there are 55 combinations of two actions including 45 different actions and 10 identical actions. These two volunteers

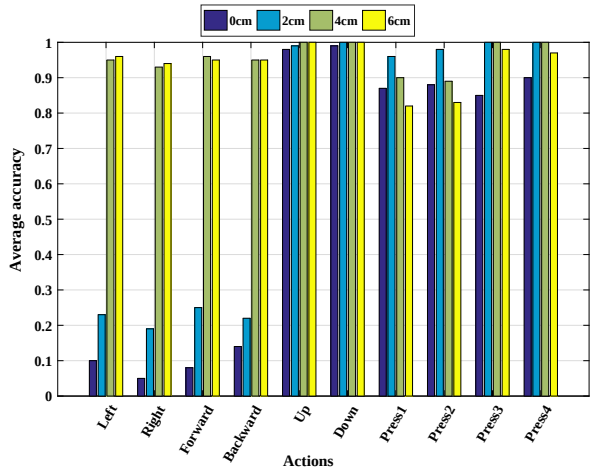


Figure 9: Impact of Tag Placement

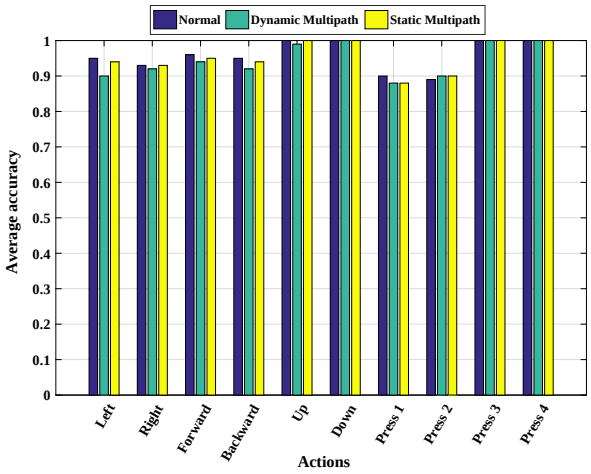


Figure 10: Effect of Resisting Multipath

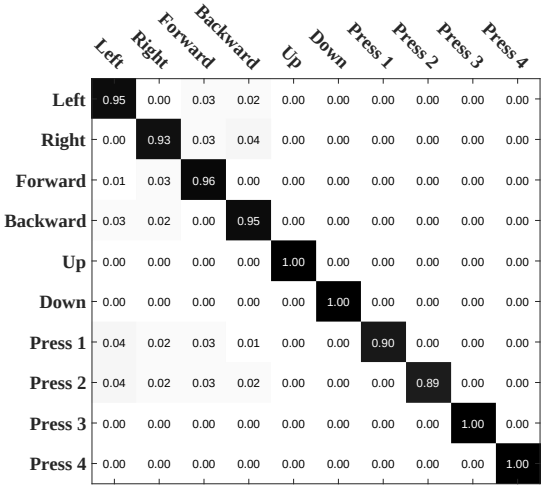


Figure 11: Effect of Removing Individual Diversity

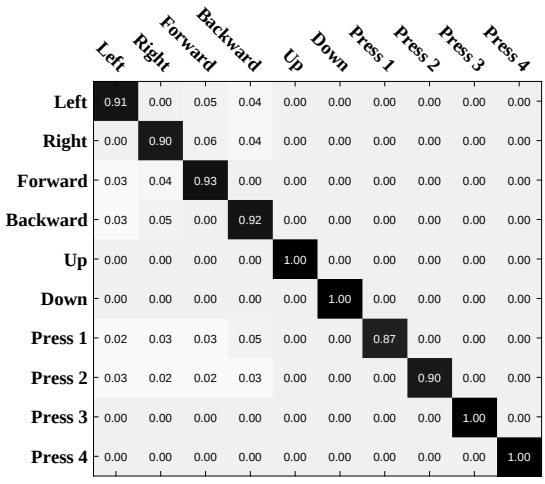


Figure 12: Feasibility of Multiple Remote Controllers

were requested to perform each combination 10 times. As illustrated in Figure 12, the average accuracy of all ten actions is 94.3%, which decreases only a little when compared with that of one remote controller shown in Figure 11 and manifests the feasibility of multiple remote controllers.

6 RELATED WORK

This section briefly reviews the related work implemented on COTS RFID devices, including novel interactive interfaces as well as activity and gesture recognition.

6.1 Innovative Interactive Interfaces

UHF RFID is normally considered as the enabling tool for automatic identification of objects. For the past few years, it has been revealing its promising applications in innovative interactive interfaces on account of its wireless, inexpensive and battery-free sensing ability. RF-IDraw [26] develops a virtual touch screen by accurately tracing the trajectory of tags with eight spatially distributed antennas. Different from RF-IDraw, PolarDraw [23] implements almost the same kind of interfaces by leveraging information from each polarization angle to infer the orientation and position of a RFID-tagged pen. Tagball [15] implements a 3D mouse which detects the motion behaviors of a set of tags by utilizing the Extended Kalman Filter. IDsense [13] provides an unobtrusive human object interaction interfaces through classifying several motion events and two kinds of touch events by Support Vector Machine (SVM). GaussRFID [14] is a hybrid RFID and magnetic-field tag sensing system that supports interactivity when embedded in retrofitted or new physical objects. PaperID [12] uses sensing and signal processing techniques to turn RFID tags into simple paper input devices and further creates a wide variety of interaction sensing types. RapID [25] develops a framework for adding low-latency input sensing

to objects through combining a probabilistic filtering layer with a montecarlo-sampling-based interaction layer. Inspired by these glittering novel interfaces, we put forward the idea of wireless and battery-free remote controllers.

6.2 Activity and Gesture Recognition

RFID based activity and gesture recognition is also a hot topic which attracts a lot of attention. The corresponding Doppler shift profile of each free-weight activity can serve as a reliable signature for each activity, thus FEMO [6] leverages this profile for on-site free-weight activity recognition and assessment. ShopMiner [24] harnesses the unique spatial-temporal correlations of time-series phase readings from backscatter signals of passive RFID tags to detect and record comprehensive shopping behaviors. GRfid [30] can recognize six device-free gestures by capturing the spatial phase features of various gestures. However, all of them are built on onerous training or prestored profiles, which is quite time-consuming and labor-tensive. Thus, we propose an action recognition system without any training or prestored profiles.

7 CONCLUSIONS

This paper proposes the first universal wireless and battery-free remote controller called TagController. In order to recognize actions without any training or prestored profiles, we elaborately devise tag placement and extract action-specific features based on careful observations and sufficient analysis. TagController is working as a pipeline which consists of three functional components, including preprocessor, FSM based action detector and action recognizer. The whole system is implemented by COTS devices and self-made remote controllers. Experimental results demonstrate that TagController can achieve an average recognition accuracy of 95.8% and 94.3% in the scenarios of one and two remote controllers, respectively. Meanwhile, TagController shows strong insensitivity to multipath surroundings, which validates the availability and robustness of our system.

ACKNOWLEDGMENTS

This research was supported by a grant from Shanghai Municipal Development & Reform Commission. Dong Wang is the corresponding author.

REFERENCES

[1] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. 2016. MagBoard: Magnetic-based Ubiquitous Homomorphic Off-the-shelf Keyboard. In *Sensing, Communication, and Networking (SECON), 2016 13th Annual IEEE International Conference on*. IEEE, 1–9.

[2] Fadel Adib and Dina Katabi. 2013. *See through walls with WiFi!* Vol. 43. ACM.

[3] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank DeRuyter. 2016. Using mobile phones to write in air. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 15–28.

[4] ALGLIB Project 2017. *Cubic spline interpolation*. ALGLIB Project. http://www.alglib.net/aboutus.php.

[5] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke recognition using wifi signals. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 90–102.

[6] Han Ding, Longfei Shangguan, Zheng Yang, Jinsong Han, Zimu Zhou, Panlong Yang, Wei Xi, and Jizhong Zhao. 2015. Femo: A platform for free-weight exercise monitoring with rfids. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 141–154.

[7] Daniel M Dobkin. 2012. *The rf in RFID: uhf RFID in practice*. Newnes.

[8] GS1 EPCglobal. 2013. *UHF Class 1 Gen 2 Standard*. http://www.gs1.org/sites/default/files/docs/uhfc1g2/uhfc1g2_2_0_0_standard_20131101.pdf

[9] Impinj, Inc. 2013. *Speedway revolution reader application note - low level user data support*. Impinj, Inc. https://support.impinj.com/hc/en-us/articles/202755318-Application-Note-Low-Level-User-Data-Support.

[10] Intel Corporation. 2017. *RealSense*. Intel Corporation. https://click.intel.com/realsense.html.

[11] Leap Motion, Inc. 2017. *Leap Motion*. Leap Motion, Inc. https://www.leapmotion.com/.

[12] Hanchuan Li, Eric Brockmeyer, Elizabeth J Carter, Josh Fromm, Scott E Hudson, Shwetak N Patel, and Alanson Sample. 2016. PaperID: A technique for drawing functional battery-free wireless interfaces on paper. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5885–5896.

[13] Hanchuan Li, Can Ye, and Alanson P Sample. 2015. IDSense: A human object interaction detection system based on passive UHF RFID. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2555–2564.

[14] Rong-Hao Liang, Han-Chih Kuo, and Bing-Yu Chen. 2016. GaussRFID: Reinventing physical toys using magnetic RFID development kits. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 4233–4237.

[15] Qiongzheng Lin, Lei Yang, Yuxin Sun, Tianci Liu, Xiang-Yang Li, and Yunhao Liu. 2015. Beyond one-dollar mouse: A battery-free device for 3d human-computer interaction via rfid tags. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 1661–1669.

[16] Tianci Liu, Lei Yang, Xiang-Yang Li, Huaiyi Huang, and Yunhao Liu. 2015. Tagbooth: Deep shopping data acquisition powered by rfid tags. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 1670–1678.

[17] Wenguang Mao, Jian He, and Lili Qiu. 2016. CAT: high-precision acoustic motion tracking. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 69–81.

[18] Microsoft Corporation. 2017. *Xbox Kinect*. Microsoft Corporation. http://www.xbox.com/en-US/xbox-one/accessories/kinect.

[19] Pavel V Nikitin, Rene Martinez, Shashi Ramamurthy, Hunter Leland, Gary Spiess, and KVS Rao. 2010. Phase based spatial identification of UHF RFID tags. In *RFID, 2010 IEEE International Conference on*. IEEE, 102–109.

[20] Kun Qian, Chenshu Wu, Zimu Zhou, Yue Zheng, Zheng Yang, and Yunhao Liu. 2017. Inferring motion direction using commodity wi-fi for interactive exergames. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 1961–1972.

[21] Jun Rekimoto. 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 113–120.

[22] Wenjie Ruan, Lina Yao, Quan Z Sheng, Nickolas Falkner, Xue Li, and Tao Gu. 2015. Tagfall: Towards unobstructive fine-grained fall detection based on uhf passive rfid tags. In *proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 140–149.

[23] Longfei Shangguan and Kyle Jamieson. 2016. Leveraging Electromagnetic Polarization in a Two-Antenna Whiteboard in the Air. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*. ACM, 443–456.

[24] Longfei Shangguan, Zimu Zhou, Xiaolong Zheng, Lei Yang, Yunhao Liu, and Jinsong Han. 2015. ShopMiner: Mining customer shopping behavior in physical clothing stores with COTS RFID devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 113–125.

[25] Andrew Spielberg, Alanson Sample, Scott E Hudson, Jennifer Mankoff, and James McCann. 2016. RapID: A framework for fabricating low-latency interactive objects with RFID tags. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5897–5908.

[26] Jue Wang, Deepak Vasisht, and Dina Katabi. 2015. RF-IDraw: virtual touch screen in the air using RF signals. *ACM SIGCOMM Computer Communication Review* 44, 4 (2015), 235–246.

[27] Teng Wei and Xinyu Zhang. 2016. Gyro in the air: tracking 3D orientation of batteryless internet-of-things. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 55–68.

[28] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. 2014. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 237–248.

[29] Lina Yao, Quan Z Sheng, Wenjie Ruan, Tao Gu, Xue Li, Nick Falkner, and Zhi Yang. 2015. Rf-care: Device-free posture recognition for elderly people using a passive rfid tag array. In *proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 120–129.

[30] Yongpan Zou, Jiang Xiao, Jinsong Han, Kaishun Wu, Yun Li, and Lionel M Ni. 2017. Grfid: A device-free rfid-based gesture recognition system. *IEEE Transactions on Mobile Computing* 16, 2 (2017), 381–393.