ChopTags: An Accurate and Low-cost Interface to Identify User/Item Interactions

Haofan Cai[1], Ge Wang[2], Josue Leyva[1], Ian Pham[1], Jinsong Han[3], Shigang Chen[4], Chen Qian[1]
[1]Department of Computer Science and Engineering, University of California Santa Cruz, USA
[2]School of Electronic and Information Engineering, Xi'an Jiaotong University, China
[3]School of Cyber Science and Technology, Zhejiang University, China
[4]Department of Computer & Information of Science & Engineering, University of Florid, USA
(email:{hcai10,joleyva,ikpham,cqian12}@ucsc.edu, gewang@xjtu.edu, hanjinsong@zju.edu.cn,sgchen8811@gmail.com)

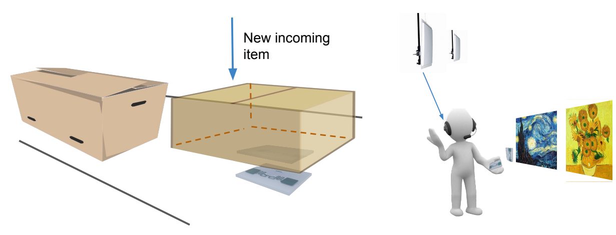
2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) | 978-1-6654-8643-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SECON55815.2022.9918168

***Abstract*—Identifying item-item and user-item interactions is an essential requirement of many ubiquitous computing applications. Recently methods of physically altering RFID tag hardware have been proposed to enable recognizing certain interactions. However, they do not address the problem that when a large number of tags exist in the environment and concurrent interactions may happen, the system may not be able to identify these interactions accurately or efficiently. We propose ChopTags, a low-cost and accurate interaction identification using passive RFID tags, with applications including automatic chess notation, shipment storage tracking, interactive libraries/retail stores/classrooms, and smart conference badges to track the attendees who had conversations. Each ChopTags module contains a passive tag chip and an antenna that are separated and can only be read when the chip is in contact with an antenna (from another pairing ChopTags module). ChopTags costs cheap hardware to scale to many users and items, achieves near 100% accuracy in complex environments, and is easy to use for children, seniors, and others who have difficulty of operating smart devices. We resolve a number of challenges of using ChopTags including improving query throughput/accuracy and identifying concurrent interactions. We build two prototypes based on ChopTags: 1) a chess auto-notation system and 2) a tag array for user interactions. ChopTags allows tracking the moves of 96 tag modules for the chess game with almost 100% accuracy and no prior work can achieve this.**

I. Introduction

Identifying item-item and user-item interactions is an essential requirement of many ubiquitous computing applications. Here "interaction" is defined as the event that a user touches an item for a particular purpose or two items touch each other to represent some relationship. Interaction identification requires the backend system of this application to recognize both the user and the item or the two items involved in the event.

Passive Radio Frequency Identification (RFID) technology is a cheap and energy-efficient solution for the Internet of Things (IoT) and has been deployed for retailing, warehousing, and transportation applications. However, identifying item-item and user-item interactions based on passive RFID tags is difficult. Despite of the current innovations of RFID sensing and localization based on analysis of physical signal features [1], [2], [5], these methods have inherent limitations to be used for large-scale and dense deployment. First, they have large errors especially in dynamic environments [3], [5], which are unacceptable for densely deployed tags. Moreover, the signal collection process may cost long time and high data volume. Last, fine-grained signal collection cannot be achieved on commodity off-the-shelf (COTS) RFID readers and need special devices such as software defined radios.



(a) Item-item application 1 (b) User-item application 1
Fig. 1. Applications of ChopTags

Recently physically altering tag hardware to achieve interaction identification has been proposed [6], [7]. However, these solutions only consider IDs as the only sensing source without exploring signal features or optimizations on RFID protocols. Hence it is unclear whether these methods can deal with concurrent interactions where multiple IDs are changing their status at the same time. *For example, our experiments shows that if there are more than 200 tags in the environment with possible concurrent interactions, more than 30% of the interaction events cannot be correctly identified using physical altering tags.*

This paper takes a low-cost and scalable approach called ChopTags, to achieve user/item interaction identification using COTS passive RFID tags. The method is the first to combine physical altering of tags and wireless signal feature analysis. We demonstrate the application of ChopTags by building a prototype called TagChess. We build a set of chess pieces and squares, each attached with a passive tag. The system can automatically record and take notations of a chess game by identifying every move with almost 100% accuracy. It avoids manual chess notations that are known to be necessary but time-consuming and distracting. In addition to the chess application, ChopTags can also be used in the following cases:

Item-item interactions. Suppose each shelf has multiple positions to place items such as shipping boxes and retail products. We attach a ChopTags module on each position as well as a module under each item. By putting all items on the shelves sequentially (Fig. 1(a)), the system can automatically record which item is put at which position. It significantly

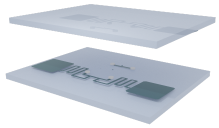


Fig. 2. A pair of ChopTags modules

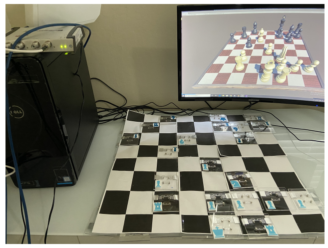


Fig. 3. The chess notation/visualization system

helps manage logistic, retailing, and manufacturing.

User-item interactions. In a museum or exhibition space, a tag is placed on the wall next to each exhibit. A visitor wears wireless headphones. She may put her tagged ticket onto the exhibit tag (Fig. 1(b)) Once the reader identifies this interaction, the backend server will stream the commentary of the exhibit to headphone, in the preferred language from the user profile. Such user-service matching requires the server knows both tag IDs. This application can be extended to amusement arcades, interactive classrooms, hospitals, cashier-free stores, and IKEA-style showrooms.

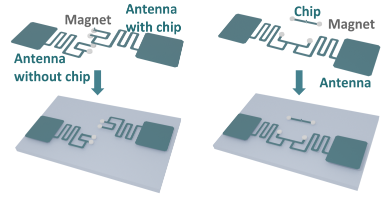
User-user interactions. Attendees of a conference may chat during a coffee break but have little time to exchange business cards. Two attendees may put their tagged badges together. The backend system will identify the two tags, exchange their e-business cards, and record the time and location. At the end of day, all related records will be emailed to each attendee, from which she can easily recall the conversations.

Our vision of such *interaction identification* is to use cheap passive tags as **ubiquitous cyber-physical connection interfaces**, which form transient links between the physical world and the cyber world. It provides an alternative tool to build future IoT systems with seamless interactions between users and smart devices.

ChopTags module (Fig.2) is implemented by a simple hardware modification of COTS passive tags motivated by recent proposals of physical altering of tag hardware [6], [7]. Each tag is disassembled into two parts: one includes the chip and the other includes the antennas. They are put into specific positions of an acrylic plate, which serves as a ChopTags module. With separation of antenna and chip, the tag cannot be powered by a reader. When two modules are snapped together, possibly with the help of neodymium magnets, both tags are connected and can be read. Tag interactions can thus be accurately identified.

ChopTags offers the following key features that make it ideal for ubiquitous applications:

- **Low-cost hardware and high scalability**: ChopTags can be made from cheap COTS tags and hence is suitable for large-scale deployment. The cost of adding more users and applications is minimal: only passive tags.
- **High accuracy**: ChopTags achieve 100% accuracy for single interaction events and $> 95\%$ accuracy for 4 pairs of concurrent interactions.
- **User-friendly**: Children, senior citizens, people with medical conditions, and others who have difficulty of using other smart devices can easily use ChopTags. ChopTags can be made into badges and bracelets.



(a) Approach 1 (b) Approach 2

Fig. 4. Tag modification approaches

We implement two ChopTags applications for case studies: one chess notation and visualization system for item-item interactions and one exhibition room for user-item interactions. We address the following main challenges in designing and developing ChopTags and its applications. 1) We present a detailed study of three different tag-modification approaches to validate the reliability of ChopTags modules. 2) We design optimized protocols that are compatible to the current EPC standard for high-accuracy and low-latency identification in complex indoor environments with many tags. 3) We present further optimization to allow ChopTags to identify concurrent interactions.

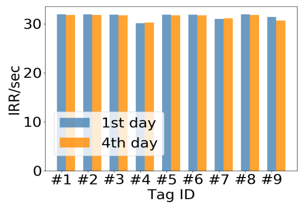
The remaining paper is organized as follows. We introduce the detailed design of the ChopTags module in Section II and show the ChopTags-based applications in Section III. We present the system implementation and evaluation results in Section IV and discuss some related work in Section V. We conclude this work in Section VI.

II. ChopTags Module Design

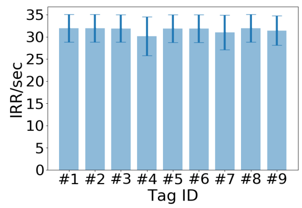
A. Modifying passive tags

As a ubiquitous interface for tag interaction identification, ChopTags should be able to satisfy the following design requirements: **R1**) reliability, **R2**) scalability, **R3**) low latency, and **R4**) ease of making. As ChopTags modules could be densely deployed in applications such as board games, which requires fine-grained detection accuracy, tag-tag interactions should be correctly identified with no error or noticeable delays. Existing methods either rely on signal features only [8]–[11] or ID information only [6], [7]. We choose to exploring both types of information in ChopTags.

Recent proposals include multiple methods of modifying tag hardware to recognize tag interactions. We studied following different approaches proposed by existing works. Note that in Fig. 4, each tag has a chip with two antennas. 1) Cutting a tag from the middle and disassembling it into two parts: an antenna and the chip with the another antenna (Fig. 4(a)); 2) Disassembling a tag into two parts: the chip and the two antennas (Fig. 4(b)) [6]. The prior studies [6], [7] only consider a single user-item interaction at a time. Their design cannot be applied directly to a densely-deployed setting with many tag modules and possibly many concurrent interations. Such a new setting presents unique challenges as we will establish in the paper. Building on the prior hardware techniques [6], we focus on the system issues that arise from multiple densely-deployed tag modules and multiple concurrent interactions. Solutions to



(a) IRR at 1st and 4th day



(b) IRR for 9 tags over 4 hour

Fig. 5. Stability performance of nine modules

these challenges will greatly expand the application scope as we have discussed in the introduction, which cannot be done by the prior art under a single-interaction design between a pair of tags.

We explore the possibility of transforming a tag into a "contact switch" by separating the chip from the antennas. We modify a tag in two ways as shown in Approach 1 (Fig. 4(a)) suggested in [9] and Approach 2 (Fig. 4(b)) suggested in [6]. Approach 1 modifies the tag into a half-half structure and disassembles it into two parts: one with an antenna and the tag chip, and the other with an antenna. Each part has two terminals affixed by neodymium magnet thereby the two parts can connect to each other. The part with the chip will act as an ungrounded monopole antenna, which performs poorly at harvesting RF energy thus making the tag not readable to the reader. When a module is interrogated by the reader, it will keep silent since the monopole antenna cannot build up enough energy. However, if another tag module is stacking on it, both tags will have complete circuits and can harvest enough power. Thus a pair of tags will be activated simultaneously. In Approach 2 the chip and the antennas of each passive tag are completely separated. Generally, neither components are readable to the reader as the chip is unable to harvest energy by itself and the antennas will not reply any ID information to the reader without an IC. When two modules are snapped together and the terminals of the chip connect with the terminals of the antennas at the proper alignment, the two tag modules can act as normal RFID tags and reply to the reader.

We first build 5 modules using Approach 1 (Fig.4(a)) and Approach 2 (Fig.4(b)) and conduct 200 interactions for each method. The reader operates in rounds. In each round it queries all tags to reply. In each trial, a volunteer picks up one of the modules and stacks it on another module. At the same time the reader keeps querying all tags. We record the *true reply rate (TRR)* of each module, defined as the probability a tag module replies its ID to reader while it is interacting with another module, and *false reply rate (FRR)* of each module, which refers to the possibility a item module replies its ID to reader while it is actually not activated by other module. We find all modules have TRRs of 100% and FRRs of 0%, meaning that a module will not reply its ID to the reader unless it is activated by another module. The results fully satisfy the requirement of this module.

We further test Approach 3 that uses a phototransistor to indicate if two tags cover each other. However, the FRR of Approach 3 is much higher than that of Approaches 1 and 2.

Stability of Contact Switches. We conduct an experiment to test the stability of ChopTags modules over a long time period. We run a 4-day experiment to keep tracking the *individual reading rate* (IRR) of each tag module, defined as the number of replies received from this tag per second. The COTS RFID reader used in our experiments support a relatively constant number of successful read operations per second (about 400). 4 modules (ID #2 $\sim$ #5) built by Approach 2 and 4 modules (ID #6 $\sim$ #9) built by Approach 2 are used in this experiment. These 8 ChopTags modules are split into 4 pairs of interacting modules (#2 with #3, #4 with #5, #6 with #7, and #8 with #9). One un-modified passive tag (ID #1) serves as the reference object for comparison. All modules are placed on a desk in an office environment. A one-hour measurement is conducted every 24 hours. During the experiments, people move and work as usually in the office. The whole measurement lasts for 4 days. Our first finding is that the IRRs of all tags on day 1 have no noticeable difference to those on day 4 (Fig. 5(a)). Fig. 5(b) presents the average IRR results collected during the four days. The average IRR of the 9 tags is about 32 replies per second with a deviation of 3.38 replies per second. All nine tags have similar IRRs, regardless of whether or not the tag has been modified. Another notable finding is that average IRR of nine tags don't change much in the four days (as shown in Fig. 5(a)). These result are important as they validate that the modified tag can stably work without performance degradation.

We eventually choose Approach 2 because we find that a module made by Approach 1 may occasionally reply ID if a user hand touches the magnet. Approach 2 has no such problem.

Remaining challenges. Simply using the ChopTags modules for user/item interaction applications is not enough. For example, in our evaluation, if there are more than 200 tags in the environment, like the case of chess, more than 30% of the interaction events cannot be correctly identified. Also it is difficult to detect concurrent interactions of multiple module pairs. In addition, the RFID protocol provides opportunities to improve the recognition efficiency when a large number of tags exist. These challenges are addressed in the next section.

III. Application-specific Designs

In this section we show the designs of two application prototypes based on the ChopTags modules by resolving the unique challenges that arise when using ChopTags. One application prototype is a chess notation and visualization system called TagChess for item-item interactions, and the other application prototype is an on-wall tag array that can simulate a museum, retailing shelf, library shelf, or smart home control panel. The challenges that need to be resolved in these two application prototypes are different. The chess prototype has to deal with a large number of ChopTags modules (256 tags in this case). Hence reducing the interaction identification latency is essential. The on-wall tag array prototype must support multiple users. Hence it is necessary to differentiate multiple pairs of modules that are simultaneously interacted.

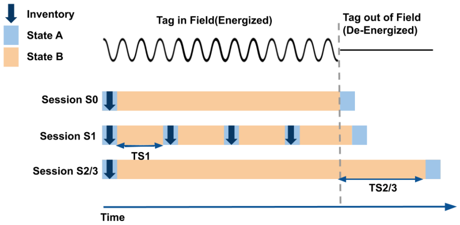


Fig. 6. Tag behaviors with different sessions

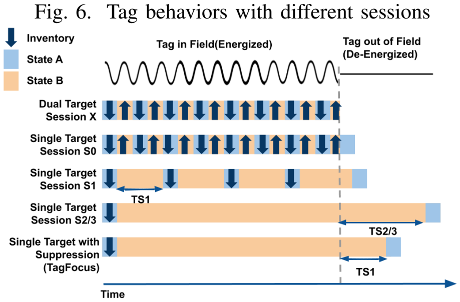


Fig. 7. Tag behaviors of different reader setups

A. TagChess for item-item interactions

Benefits of TagChess. In this TagChess prototype, all chess pieces and squares on the chessboard are augmented with ChopTags modules, which we refer as *piece modules* and *square modules* respectively. Using TagChess, all moves will be recorded in a database and visualized on a computer screen. Hence notations are automatically taken. It is well-known that manual chess notations by the players are necessary but time-consuming and distracting. In addition, it provides an interface that allows one player to use a real chess board to play with an Internet opponent.

In the initial position, 64 modules (32 piece modules and 32 square modules) are activated and 32 other modules are not. The system needs to monitor their status changes throughout the game. We define the interaction event of moving a piece module from one square module to another one as a **PieceMove** event. Each piece also has two statuses: *'on-board'* and *'off-board'*, indicating whether the piece module is being activated or not. An on-board piece is a piece that is steady on a square, and an off-board piece can either be a captured piece or one held by the player in the air during a move.

Intuitively, if a piece module is moved, its status switches from 'on-board' to 'off-board', then back to 'on-board'. Current RFID protocols uses slotted ALOHA as the MAC layer solution, which requires tags to compete for time slots to reply to the reader. The commodity RFID reader used in our experiments support at most 400 read operations per second, regardless of the number of tags and the communication distance. When the number of tags is 64, the IRR of each on-board piece module is about $5 \sim 7$ reads/sec. The IRR of an off-board piece module will maintain 0 reads/sec until this piece module is placed back to a square module. However, if we try to leverage the IRR change as an indicator to infer the status change of a piece module, we will receive a detection result with large error, as signal blockage and environmental dynamics sometimes may cause the decrease in a piece's IRR. In order to achieve high accuracy, status changes of a piece module must be continuously and precisely detected.

1) EPC Protocol-based Method Design.: In TagChess we want a piece module to reply its ID if and only if it has been moved, and keep silent until the next move. Hence we explore the inventorying features of passive RFID tags and try to make the tags 'behave' as we expect. COTS RFID readers can be configured using a number of settings to handle different use cases. In our design, we mainly focus on two important configuration settings: *Session* and *Search Mode*, to improve the identification accuracy.

The EPCglobal C1G2 air protocol specification [12] provides a mechanism called "Session" to allow multiple readers to communicate with a single tag. According to the standard, a reader can send out queries to the tags in one of four sessions (denoted as $S0$, $S1$, $S2$, and $S3$). Hence at most four readers can queries the same tag in parallel. For each session, a tag may be in two states represented by an *inventoried flag*, namely state A and state B. If a tag's initial state is A in a session and it is queried and replies to the reader querying this session, its state switches to B and vice versa. A reader can select to query only state A tags or state B tags. By allowing the reader to only query state A tags, we can avoid a tag to reply to the reader for multiple times. However in practical situations a tag with state B may also switch back to A, and the situations differ for the four different sessions. For the rest of this paper, we say a tag "in field" if it is within the interrogation range of the reader and its chip and antennas can be powered, thus it can receive the signals from the reader. And we say a tag "out of field" if it cannot receive the reader signal due to either being out of the interrogation range or being a single ChopTags module.

The initial value of the inventoried flag depends on the session used by the reader as well as the previous value of the inventoried flag. Each session has an associated *persistence time*, which is the time duration that the inventoried flag is maintained after the reader stops energizing the tag. Detailed tag behaviors are discussed as followed (shown in Fig. 6):

- $S0$: When a tag is in field, the inventoried flag of $S0$ is always set to A. It has zero time for persistence, indicating that when a tag is out of field and not energized by the reader, its inventoried flag will immediately return to A.
- $S1$: $S1$ inventoried flag can have initial value of either A or B, depending on its previous state. The inventoried flag is set to previous value (either A or B) if the tag is energized before the persistence time expires, otherwise the inventoried flag will default to A. For instance, when it is in state A and is queried by the reader, the state switches to B. State B will last for a persistence time $T1$ from 0.5 to 5 seconds. After the persistence time expires, the inventoried flag will revert to A, no matter whether the tag is in field or out of field.
- $S2$ and $S3$: The initial value of $S2$ or $S3$ inventoried flag is set in the same way as session $S1$. If a tag is in state A, it can be queried by the reader and can reply to the reader. Then the state switches to B. Its state will

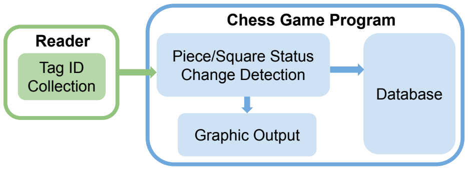


Fig. 8. Overview of the chess game system

keep in B as long as it is in field. Even if it is out of field, the state will still be B until a persistent time $T2/3$ expires. Note $T2/3$ can be very long(with a minimum of 2 seconds).

Our design requirement is that if a ChopTags is snapped with another module and being in field, it should reply once and become silent. When it is separated with another module (out of field) and snapped with another new module (in field again), it should be able to reply again. From that requirement, none of the above sessions is ideal. $S0$ definitely does not meet the requirement because tags cannot be silent. $S1$ will allow the tags to reply again after $T1$. For $S2/S3$, it is nice that a tag only replies once when it is in field. However, when it becomes out of field, we need to wait a long time $T2$ to make it able to reply again. We cannot assume a player should wait that long time until a move is identified. We further investigate other settings of RFID system.

We share the following *experimental observations* from our tests with a ImpinJ Speedway modeled R420 RFID reader, which is one of the most commonly used COTS RFID reader devices in research [6], [11], [13], [14], as shown in Fig. 7. Besides "Session", Impinj RFID readers also provide another important mechanism called "Search Mode" that will also influence a tag's overall behavior, hence can be configured to handle various use-cases. Impinj software and libraries allow users to change these settings. There are five search modes available on the Impinj reader used by us: (1) Dual Target; (2) Single Target; (3) Single Target with Suppression (TagFocus); (4) Dual Target Select $B \rightarrow A$; (5) Single Target Reset. Note that "Target" refers to whether the reader will only query tags that are in a single inventoried flag, either A or B (Single Target), or it will interrogate tags regardless of their inventoried flags (Dual Target). Note that due to channel hopping in actual implementation, for Single Target with session $S0$, the tag will behave similarly to Dual Target, as shown in Fig. 6.

We find that we set TagFocus with session $S1$, a tag only replies when it is in field, and the persistence time is $T1$, which is much shorter than $T2$ in practical situation. Hence it satisfies our requirement of ChopTags.

Therefore, we adopt the TagFocus setting in ChopTags, by which the tag will only reply for one time after being activated, and remain quiet until it has been unpowered. A piece module and its interacting square module will reply only once respectively and then keep silent, until the piece is moved. When this piece module is placed on a new square module, the newly formed piece-square interacting pair will be activated at the same time, thus again reply their IDs for once and remain silent.

2) Algorithmic optimization: The overview design of TagChess is shown in Fig. 8. It uses the data collected from the RFID reader to generate high-level interaction events (e.g., PieceMove in a chess game) by querying the backend database, and the final result will be shown as the graphic output in the computer screen. Further algorithmic optimizations are proposed to handle the status change detection process.

Two sets of modules, namely the piece module set (P) and the square module set (S), are pre-registered in ChopTags system. P includes all 34 tag IDs of the pieces – assuming each player has one extra backup queen – and S includes all 64 IDs of the squares. For each module in S, its corresponding row and column numbers are stored in the database.

When the RFID reader begins querying, by adopting the aforementioned TagFocus configuration, all the activated squares and pieces will remain quiet after replying their IDs once. After that, each piece will reply its ID only if a new PieceMove event is triggered, implying that the piece has been moved from the original square and reconnect to another square. The system obtain a list H of **moved** tag ID, which refers to the module who has just experienced a state change. Reader is set to report interaction events once a second. The total number of moved tags is recorded as $|H_n|$. In TagFocus configuration, if piece p_1 is moved from square s_1 to square s_2, then only modules of p_1 and s_2 will report status changes and s_1 will be unpowered. A *capture* move is composed of two sub-steps: 1) the captured piece module is removed from its square and 2) the capturer is moved from its original square to the square of the captured piece. Step 1) will not generate status changes and step 2) will report two status changes. All special cases, including *castling*, *en passant*, and *promotion*, are properly handled and can be successfully identified by TagChess. All other updates are regarded as invalid ones and a warning sign will be shown on the screen.

The $|H| > 2$ cases are mainly caused by moving multiple pieces and stacking them on different squares at the same time, which is invalid in the chess game – even *castling* and *en passant* should be moved in an order. Although this case seldom occurs in item-item interactions, it may frequently occur in other application scenarios when users' synchronous interactions are allowed. We will address this problem and propose an effective solution in the following section.

B. Tag array for user-item interactions

In the applications of the museum auto-commentary system discussed in Section I, smart retailing shelf, library shelf, and smart home control panel, the interaction pattern is that multiple tag modules are placed in the environment to represent different objects (called the *item module*) and each user holds a tag module (called the *user module*) to interact with the an item module. The backend system should identify the interacting modules, record these interactions, and provide further feedback (e.g., sending the museum commentary to the user's headphones or turning on a certain appliance.)

The major challenges of this type of applications is that multiple users may interacting with different item modules at the same time. Hence the system may receive four (or more) simultaneous status changes from four tags and needs to solve the module pairing problem, i.e., deciding which module is interacting with which. Intuitively, two interacting tags should be in close-proximity and hence have similar RF channel features.Thus we leverage the **channel similarity** in RFID communication to resolve the module pairing problem. Channel similarity is an important observation from practical RFID communication: If two tags are put in a physical proximity (e.g., 2cm), they will show high similarity on their received phase information [3].

1) Channel Similarity: Channel similarity is based on the fact that nearby RFID tags experience a similar multipath environment (e.g., reflectors in the environment) and hence exhibit similar multipath profiles. Existing works [3], [11], [15] demonstrate that the channel conditions are extremely similar for two tags that are in close proximity We propose to use the received tag phase information to determine the similarity. For simplicity, we use tag T_i to refer to the ChopTags module built with tag T_i in the following content.

In an RF environment, the tag T_i's phase changes over the line-of-sight (LOS) distance can be calculated as:

$$\phi_{d_{ij}} = 2\pi(\frac{2d_{ij}}{\lambda}) \mod (2\pi), \tag{1}$$

where d_{ij} is the LOS distance between tag T_i and antenna A_j. Besides the dominated phase changes over distance, the received phase ϕ_{ij} of tag T_i at antenna A_j also comprises of the initial phase of the tag and the antenna, i.e., ϕ_{T_i} and ϕ_{A_j}, respectively. We can represent ϕ_{ij} as follows:

$$\phi_{ij} = (\phi_{d_{ij}} + \phi_{m_{ij}} + \phi_{T_i} + \phi_{A_j}) \mod 2\pi, \tag{2}$$

where $\phi_{m_{ij}}$ represents the phase changes introduced by the multi-path effects. However, it can be noted from Eq. 2 that even if two different tags are placed in close physical proximity, i.e., they have similar ϕ_d and ϕ_m, their measured phases at the same antenna are still highly likely to be different since the received phases are also impacted by the initial phases ϕ_{T_i} and ϕ_{A_j}, which varies by hardware diversity.

To deal with the phase deviations introduced by device-dependent features, we use **differential sensing** in [11], i.e., deploy multiple antennas in our system and measure the difference between tag's phase values received by multiple antennas, instead of measuring a tag's absolute phase value from a single antenna. The intuition underlying this design is that even though different tags have ambiguous and diverse initial phases, such difference can be canceled using the measurement from two antennas.

After adopting 2 antennas in the system, the difference of the phases of tag T_i and T_j that collected at antennas A_1 and A_2 can be calculated as:

$$\begin{aligned} &\Delta\phi_i = \phi_{i1} - \phi_{i2} = (\phi_{d_{i1}} - \phi_{d_{i2}}) + (\phi_{m_{i1}} - \phi_{m_{i2}}) + \\ &(\phi_{A_1} - \phi_{A_2}). \\ &\Delta\phi_j = \phi_{j1} - \phi_{j2} = (\phi_{d_{j1}} - \phi_{d_{j2}}) + (\phi_{m_{j1}} - \phi_{m_{j2}}) + \\ &(\phi_{A_1} - \phi_{A_2}). \end{aligned} \tag{3}$$

where $\phi_{A_1} - \phi_{A_2}$ is a constant. Therefore when tag T_j and tag T_i are close, they experience quite similar LOS path (as the path difference between them is much less than the LOS propagation path) as well as multipath changes, implying that they should have similar values between $\phi_{d_{i1}} - \phi_{d_{i2}}$ and $\phi_{d_{j1}} - \phi_{d_{j2}}$, as well as $\phi_{m_{i1}} - \phi_{m_{i2}}$ and $\phi_{m_{j1}} - \phi_{m_{j2}}$. Hence $\Delta\phi_i - \Delta\phi_j$ should always be a small value for samples at any time point. As a result, the phase difference $\Delta\phi_i$ of tag T_i should be close to that of tag T_j, $\Delta\phi_j$. While for those tags that are placed far away from tag T_i, they are likely to have different LOS distances as well as multipath environments with tag T_i, and hence their phase difference would have a much larger gap with that of $\Delta\phi_i$. We define the **phase difference profile** D_i for tag T_i as a vector where each element is $\Delta\phi_i(t)$ at time t. By leveraging the phase differences, we can further determine the interacting tag pair among all activated tags.

2) Interaction Pairing Details: If tag T_i and T_j are the pair of modules interacting with each other, D_i and D_j should have similar values, i.e., the Euclidean distance between these two vectors should be small. We use the distance metric $Dist(D_i, D_u)$ to refer to the Euclidean distance between D_i and D_j and use this metric to evaluate the similarity between phase difference profiles.

Suppose n user modules and m item modules are simultaneously activated (sometimes $n \neq m$ as the module may not be correctly activated). When the reader has finished collecting the data for a time duration T, it immediately begins measuring the similarities between phase difference profiles over these $n + m$ modules. The choice of T determines the latency and processing overhead of computation. In our final implementation of ChopTags, $T = 2.5$s as we have empirically determined it to be suitable for a practical deployment. We have also evaluated the performance of ChopTags with other values of T and the results will be presented in Section IV-C.

The basic approach is to match the measured phase difference profiles of all item modules against the profiles of the user modules, aiming to find user-item pairs that have the smallest Euclidean distance. ChopTags first gets phase difference profiles $D_U^1, D_U^2 ... D_U^n$ for n user modules as well as $D_I^1, D_I^2 ... D_I^m$ for m item modules, These $n + m$ vectors should have equal length L after interpolation. ChopTags then iteratively calculates the Euclidean distance between the phase difference profile vectors of one specific user module u and m item modules,

$$E_u^i = \sqrt{\sum_{k=1}^{L} (d_k^i - d_k^u)^2} \tag{4}$$

where d_k^i and d_k^u are the k-th elements of the phase difference profile D_I^i and D_U^u, respectively. ChopTags performs calculations between user module u and n item modules, and then finds the item module i who has the least E^i values that satisfy $|E_u^i| \leq E_h$, which are most likely to be paired with user module u. $E_h = \lambda\sqrt{L}$ where λ is a empirically pre-defined threshold for comparison. Notice that under some circumstances, for a specific user tag u, none of the item tags satisfy the $|E_u^i| \leq E_h$ requirement due to mis-manipulation. Hence no matching result is provided for user module u. In

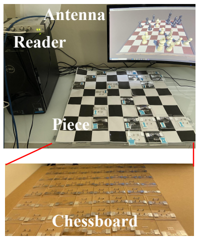


Fig. 9. Prototype environment

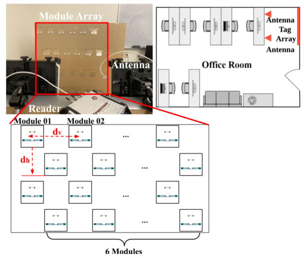


Fig. 10. Deployment of the tag array in an office

practice the user can easily solve this by retrying interacting with the item module.

IV. Implementation and Evaluation

A. ChopTags Module Implementation

The system prototypes are implemented based on COTS UHF RFID devices: an ImpinJ Speedway modeled R420 RFID reader, two Laird S9028-PCL directional antennas, and ChopTags modules built with three types of UHF RFID tags: AZ-9662, ImpinJ E41C/B, and Alien 964X. The chip and antenna of each tag was disassembled into two components, as shown in Fig. 4(b). All these prototypes are compatible to the standard EPCglobal Class 1 Generation 2 protocols (C1G2). In fact, even if two ChopTags modules are built with different model of tags, interaction event can still be identified by the reader and the backend system. In the following experiments, we only present the evaluation results of ChopTags modules built with AZ-9662 tags. One COTS RFID reader is sufficient to cover a large indoor area as it can connect to multiple antennas and the antennas do not have to be placed at the same location of the reader. A wire cable can be used to connect an antenna to a reader. The transmission and receiving gains are both set to be 32.5dB. We run the software components of the prototypes at a Dell desktop with Intel Core i7-7700 CPU at 3.6GHz and 16G memory.

This work focuses on the applications in indoor environments such as museums, schools, and homes. These environments usually contain various multipath reflectors, moving objects, electrical devices, as well as wireless signals. We evaluate the prototypes of ChopTags in an typical office room (as shown in Fig. 9).

For each ChopTags module, the tag components are manually placed on a 70 (W) $\times$ 70 (L) $\times$ 20 (T) mm^3 laser-cut acrylic plate (Fig. 11). Two neodymium magnets with 4 mm radius and 2 mm thick are fixed to the terminals of each component as the magnetic connectors. The distance between IC component and antenna is around 20 mm (may vary during manufacture). Magnets attached on IC component and antenna part have exact opposite polarity facing upward to ensure snapping connection. The manual installation of each unit can be fabricated in 3-5 minutes by a graduate student. Note we do not intend to choose the best material or size of the module, which for sure have a big room to improve.

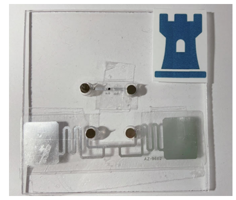


Fig. 11. A ChopTags module of rook

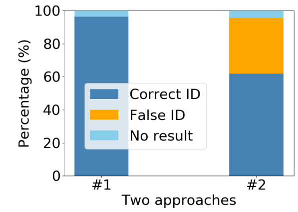


Fig. 12. Accuracy of two approaches

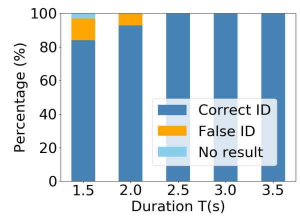


Fig. 16. Impact of signal collection time T

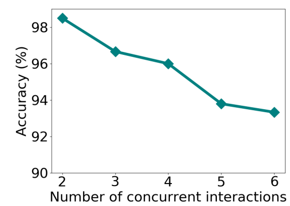


Fig. 17. Accuracy of concurrent interactions

B. TagChess Evaluation

1) Methodology: Fig. 9 shows an appearance of the TagChess prototype. The experiments are conducted on a platform in the aforementioned office room. A 8×8 grid of square modules are deployed on the platform as a chess board, while piece modules are stacking on the chess board that allow users to play with. A reader antenna is hanging above the platform to monitor the interaction events. The distance between the reader antenna and the chess board is 0.8m. The graphic program of TagChess is developed in Unity3D.

To better evaluate the performance, we compare our EPC protocol-based design (which is referred as 'Method 1') with the vanilla design ('Method 2') leveraging IRR changes as the indicator to infer the status change of a module. By using Method 2, the system should regard a module (piece/square) as experiencing a PieceMove event if the IRR increment of this module in two consecutive seconds is larger than a threshold T, while at the same time its IRR in the first second is close to 0 reads/sec. System will update and report the ID every 2s. The threshold T is dynamically refreshed to be half of the average IRR of all piece modules in the last update.

2) Performance.: We first show the latency of TagChess. The response time mainly consists of four parts, namely time for data (replies) collection, time for signal processing, time for generating graphic output in Unity3D, and the message exchanging duration between reader and server programs. We start by placing 32 piece modules on the chess board, and then move the these pieces for 50 moves, and obtain the average refresh time for these four sections. The results are present in Table I. As can be seen from the result, the latency is mainly contributed by data collection, which is set to 1 second (Method 1) and 2 seconds (Method 2). Method 2 requires significantly long time. Later we show that even with 2x data collection time, Method 2 still cannot identify all moves correctly.

We define three types of identification for each PieceMove event, *correct reports*, *false report*, and *no result*, to evaluate

Section	Data Collection	Signal Processing	Graph Processing	Message Exchange between Applications
Method 1	1 sec	5.04 ms	11.20 ms	2.82 ms
Method 2	2 sec	7.43 ms	11.20 ms	2.90 ms

TABLE I
LATENCY OF TAGCHESS

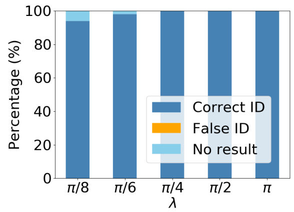


Fig. 13. Impact of threshold λ

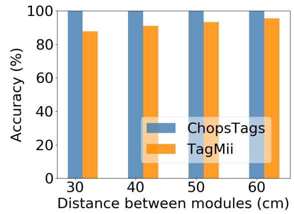


Fig. 14. Impact of distance bewteen modules (sparse deployment)

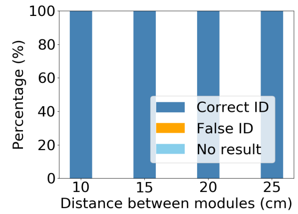


Fig. 15. Impact of distance between tags(dense deployment)

the accuracy of TagChess. Correct reports are those events identified correctly by TagChess. False report are those events identified incorrectly by TagChess. Sometimes user may not be able to see the correct move she just operated due to some mismanipulation. She can retry connecting the piece and square modules. However, if no ID is reported after trying for three times, then we consider this PieceMove event receive a 'No result' feedback from the system .

We conduct 250 moves and compute the accuracy of TagChess. As shown in Fig. 12, the protocol-based design (Method 1) is very reliable. All events are correctly identified. While for method 2, it will produce 30%+ wrong pairing results. One reason is that the environmental dynamics may also cause the increase of a module's IRR.

C. Tag Array Evaluation

1) Methodology: We evaluate the performance of the ChopTags based tag array system in a complex office environment. We use a module array with 4 rows and 7 columns as shown Fig. 10 to simulate the modules attached on a wall of a museum or items on a shelf. The distance between the reader antennas and ChopTags's module array is 1.2m in the office room. Before the experiment, the horizontal distance d_h between two adjacent module is initially set to be 25cm, while the vertical distance d_v between them are set to be 30cm. In our experiments, we invite 4 volunteers with heights varying from 158cm to 177cm. We let the volunteers randomly move in the testing space and use their ChopTags modules to interact with the item module in the space. Every accuracy value shown in this section is the average of 100 interaction experiments. We compare the results with the recently proposed TagMii [11]. Note the results of TagMii are directly got from the original paper [11].

We first evaluate the performance of ChopTags in the situation where there is only one user interacting with the an item at a same time, in the dynamic environment where multiple users move around. Accuracy μ is defined as $\mu = n_c/n_0$, where n_0 is the total number of interaction events operated by the user, while n_c is the number of events where the IDs of involved pair of ChopTags modules are both correctly identified. The false report rate ϵ is defined as $\epsilon = n_e/n_0$, where n_e is the number of interaction events reported, which can be either reporting a wrong user module ID or a wrong item module ID.

Impact of parameter λ: The comparison threshold $E_h = \lambda\sqrt{L}$ in Section III-B2 plays an important role in judging whether an item module can be considered as an interacting target with the user module. A small λ may cause the miss of target item module. We vary λ from $\frac{\pi}{8}$ to π, and show the accuracy μ of ChopTags in Fig. 13. ChopTags can maintain an accuracy of about 100% when $\lambda = \frac{\pi}{4}$ and this value is consistent across different setups, hence we set the λ to be $\frac{\pi}{4}$.

Accuracy for non-concurrent interactions: We conduct the experiments by varying the horizontal distance between adjacent item modules from 10cm to 60cm in the office. To better evaluate the performance of ChopTags as well as compare with the previous work, we divide the experiments into two categories, namely dense deployment where 10cm$\leq d_h \leq$ 30cm, and sparse deployment where 30cm$< d_h \leq$ 60cm. For sparse deployment we compare the evaluation results with those in TagMii which have similar deployment settings of Fig. 14. TagMii cannot deal with dense deployment with distance $<$ 30cm. As illustrated in Fig. 15 and Fig. 14 , for the single user case, even if the deployment of module array is much denser (d_h = 10cm) compared to TagMii, the accuracy is always 100%.

A main contribution of ChopTags is that it can support and resolve multiple concurrent interactions. We first study the case when two users interact with their target item modules at the same time, and gradually increase the number of users to demonstrate how the number of users will influence the system performance.

Signal collection duration T: As mentioned in Section III-B2, ChopTags estimates the similarities after every time interval T. Intuitively, a longer data collection duration will result in higher accuracy. However, the user experience will be greatly downgraded if the duration lasts for too long. We vary the time duration T from 1.5s to 3.5s with an interval of 0.5s and evaluate the accuracy in Fig. 16. From the results, when $T = 2.5s$, ChopTags can achieve an accuracy of 100% of pairing and identification. Thus, we require each user to put the user tag for at least 2.5s in the experiment before receiving the feedback.

Impact of number of concurrent interaction events per antenna: We further study the case when more than two users concurrently interact with their target item modules. Note that the d_h is still initially set to be 25cm, and the vertical distance d_v is set to be 30cm. During the experiments, we ask the volunteers to arbitrarily interact with their target modules at the same time without caring about the distance between two adjacent activated item modules. Fig. 17 presents the experiment accuracy as we gradually increase the number of concurrent interactions. When less than 4 users per antenna interact with the item modules at the same time, ChopTags system can still maintain a high accuracy above 96% with generating a small fraction of wrong results.

V. Related Works

RFID sensing uses physical signal features from tags to detect their status, which is a promising approach for passive tag based applications such as gesture-based inputs [1], [2]. IDSense [8] enables coarse-grained touch events recognition of objects using real-time classification of PHY-layer signal features. PaperID [9] is a similar work that uses supervised machine learning to detect different types of on-tag and free-air interactions with custom-designed RFID tags. Pradhan *et al.* [10] show how changes in the received signal phase caused by touching on RFID tag can be leveraged to detect the finger swipe or touch gesture without any pre-training stage. To our knowledge, the only work that addresses a similar problem to ChopTags is the recently proposed TagMii [11]. However, TagMii relies on similar signal features to identify tag interactions. When the number of background tags increases, the identification accuracy might be lower than 80%. All tags are required to be at least 30cm apart, which significantly limits the applications. In addition TagMii requires around 10s to collect enough signals for each interaction. All these solutions require a considerable time and computation for signal analysis, while their performance is also affected by the environmental dynamics.

Recent studies have proposed to physically modify the tag circuitry and detect the correlated state changes of electromechanical sensors in the tag circuitry caused by human behaviors [6], [14]. RFIBricks [6] is a building-block system which symmetric 2D patterns of RFID contact switches are deployed on the top and bottom of each block. Such design enables the backend system to recognize which block is stacked on which, as well as the stacking orientation. RFIMatch [14] detects finger-touching on a tag based on the correlated state change between the tag and an RFIMatch fingerstall worn by user. Different from ChopTags, these methods are not a generalized approach to identify tag interactions and they do not solve the problems such as simultaneous interactions and long query time of large-scale tags.

VI. Conclusion

ChopTags is a novel solution for tag interaction identification developed with COTS passive RFID tags, which may enable many ubiquitous computing applications that require the recognition of user/item interactions. ChopTags is the first to combine the information of both tag ID presence and physical signal features to infer interactions. ChopTags achieves near 100% accuracy and only requires every user or item to carry a passive tag. We implement two application prototypes based on ChopTags and evaluate the prototypes in complex environments.The results how that ChopTags is highly accurate and reliable with low latency.

VII. Acknowledgement

H. Cai, J. Leyva, I. Phan, and C. Qian were partially supported by National Science Foundation Grants 1750704, 1932447, and 2114113. G. Wang is supported by National Natural Science Foundation of China Grant 62002284 and J. Han is supported by National Natural Science Foundation of China Grants U21A20462, 61872285. We thank the anonymous reviewers for their suggestions and comments.

References

[1] P. Asadzadeh, L. Kulik, and E. Tanin, "Gesture recognition using rfid technology," *Personal and Ubiquitous Computing*, vol. 16, no. 3, pp. 225–234, 2012.

[2] J. Wang, D. Vasisht, and D. Katabi, "Rf-idraw: virtual touch screen in the air using rf signals," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 235–246.

[3] J. Wang and D. Katabi, "Dude, where's my card?: Rfid positioning that works with multipath and non-line of sight," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 51–62.

[4] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proceedings of ACM MobiCom*. ACM, 2014, pp. 237–248.

[5] G. Wang *et al.*, "An Universal Method to Combat Multipaths for RFID Sensing," in *Proc. of IEEE INFOCOM*, 2020.

[6] M.-J. Hsieh, R.-H. Liang, D.-Y. Huang, J.-Y. Ke, and B.-Y. Chen, "Rfibricks: interactive building blocks based on rfid," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 189.

[7] J. Wang, O. Abari, and S. Keshav, "Challenge: RFID Hacking for Fun and Profit," in *Proceedings of ACM MobiCom*. ACM, 2018, pp. 461–470.

[8] H. Li, C. Ye, and A. P. Sample, "Idsense: A human object interaction detection system based on passive uhf rfid," in *Proceedings of ACM CHI*. ACM, 2015, pp. 2555–2564.

[9] H. Li, E. Brockmeyer, E. J. Carter, J. Fromm, S. E. Hudson, S. N. Patel, and A. Sample, "Paperid: A technique for drawing functional battery-free wireless interfaces on paper," in *Proceedings of ACM CHI*. ACM, 2016, pp. 5885–5896.

[10] S. Pradhan, E. Chai, K. Sundaresan, L. Qiu, M. A. Khojastepour, and S. Rangarajan, "RIO: A Pervasive RFID-based Touch Gesture Interface," in *Proceedings of ACM MobiCom*. ACM, 2017, pp. 261–274.

[11] H. Cai, G. Wang, X. Shi, J. Xie, M. Wang, and C. Qian, "When tags 'read' each other: Enabling low-cost and convenient tag mutual identification," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 2019, pp. 1–11.

[12] E. Global, "Specification for RFID Air Interface EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz," Technical report, GS1, Tech. Rep., 2008.

[13] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu, "Relative Localization of RFID Tags Using Spatial-Temporal Phase Profiling." in *NSDI*, 2015, pp. 251–263.

[14] R.-H. Liang, M.-J. Hsieh, J.-Y. Ke, J.-L. Guo, and B.-Y. Chen, "Rfimatch: distributed batteryless near-field identification using rfid-tagged magnet-biased reed switches," in *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 2018, pp. 473–483.

[15] G. Wang, C. Qian, J. Han, W. Xi, H. Ding, Z. Jiang, and J. Zhao, "Verifiable smart packaging with passive RFID," in *Proceedings of ACM UBICOMP*, 2016.