HMRL: Relative Localization of RFID Tags with Static Devices

Ge Wang[†], Chen Qian[*], Longfei Shangguan[‡], Han Ding[†], Jinsong Han[†], Nan Yang[†], Wei Xi[†], Jizhong Zhao[†]
[†]Xi'an Jiaotong University, China
[*] University of California, Santa Cruz, USA
[‡] Princeton University, USA

***Abstract*—Passive Radio Frequency Identification (RFID) tags have been widely applied in many applications, such as logistics, retailing, and warehousing. In many situations the relative locations of objects are more important than their absolute locations. However, state-of-art relative localization methods need continuing movement of tags and readers, which limit the application domain and scalability. In this paper, we propose a relative localization approach for passive tags that requires no device movement. Instead, our method utilizes signal changes caused by arbitrary movement of human beings around tags, who carry no device. Hence our method is called Human Movement based Relative Localization (HMRL). The basic idea of HMRL is that when people pass between reader antenna and tags, the received signal strength will change. By observing the time-series RSS changes of tags, HMRL can obtain the order of tags along a specific horizontal direction. HMRL can also get the order of tags in a vertical direction using hyperbolic positioning. We implement HMRL with commodity off-the-shelf RFID devices. The experimental results show that HMRL achieves high accuracy for relative localization of passive tags.**

***Index Terms*—RFID, Relative localization.**

I. Introduction

Passive Radio Frequency Identification (RFID) tags have been widely applied in many applicationssuch as logistics, retailing, and warehousing. Besides identification, object locations have become increasingly important in those applications. Hence a number of solutions have been proposed to localize passive tags [13][6][10][5]. However, most of them require expensive devices and infrastructure (*i.e.*, multiple antennas [5] or synthetic aperture radar(SAR) [10]), or continuous tag movement [13]. These constraints limit the applications of tag localization.

Very recently, researchers have identified the importance of relative localization of RFID tags [7][8]. Instead of reporting the physical locations of each tag (called absolute localization), relative localization tells the relative locations of all tags. For example, in a one-dimensional space such as a moving conveyor belt, for any two tags a and b, relative localization must tell whether a is ahead of b or in reverse. In a two-dimensional space such as a shelf, relative localization must tell the relative location of an arbitrary pair of tags a and b, such as "a is below b vertically and to the left of b horizontally". Compared to absolute localization, relative localization is simple and significantly saves the infrastructure cost, when absolute locations are not necessary in the application. We identify the following user cases for relative localization.

- In a library, correct orders of the books or archival brochures are essential for readers to find their targets using the library call numbers. Managers can use relative localization to ensure that the books are placed in the right order.
- A large number of boxes (or baggages) are placed in an order to be loaded to a truck (or cargo aircraft). The truck will unload the boxes to multiple destinations hence we need ensure that the boxes that are unloaded earlier should be placed closer to the gate. In this application the absolute locations of the boxes are not important but the orders should be identified and checked.
- In a retail store, the manager needs to ensure that items in a shelf is correctly placed and consistent to the labels. As long as the manager knows the relative locations of the items are correct and items are tightly placed, s/he can conclude that the placement is good. It is because the shelf and items all have fixed size. If the orders are correct, miss-placement can hardly happen.

However, prior relative localization approaches still have two rigid requirements (or limitations) similar to absolute localization solutions: 1) the continuous movement of the reader or tags, and 2) avoiding the impact of environment noises (*e.g.*, human movements). For example, OTrack [7] is designed only for tracking the order of objects on a moving conveyor belt. STPP [8] recognizes the relative positions of tags by observing the sudden changes of phase values from different tags when the reader antenna moves along a known direction. If there are people moving nearby or between the reader and tags, these movements can also incur sharp changes on the phase values. As a result, the recognized relative locations are extremely inaccurate.

Unfortunately, these two requirements are hardly to satisfy in practice. Except for some specific cargo systems that are with moving conveyor belts, objects as well as their tags always keep static in most time, such as the books or other items tightly placed in the shelves. Under these circumstances, moving either the reader or tags is impractical, considering the dedicated space for accommodating the moving devices or infrastructures. Holding portal RFID readers to scan all tags may avoid the interference caused by moving objects. But this solution should be manually performed and hence increases the infrastructure and labor cost. In addition, it cannot provide real-time localization.

978-1-5090-6599-8/17/$31.00 ©2017 IEEE

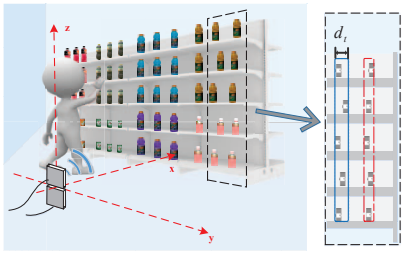


Fig. 1. Overview of HMRL

In this paper, we propose a Human Movement based Relative Localization system, namely HMRL, using passive RFID to achieve accurate relative localization. HMRL leverages the arbitrary movement of human beings in the region, rather than devices, to explore the spatial information about the tags. With the help of human movement, HMRL can acquire two-dimensional relative locations of passive tags on a shelf or rack as shown in Fig. 1. Nevertheless, we should address the following challenges in the design and implementation of a practical HMRL system.

- **Arbitrary human movement**. Human movement in the region is uncontrollable. It introduces non-trivial uncertainty and increases the difficulty in analyzing signal changes of tags.
- **Device diversity**. Due to the imperfection of manufacture, different tags exhibit significant differences in their signal features.
- **Multiple-person impact**. In real-world applications, it is quite common that multiple persons walk within the area of interests. When they move simultaneously, the line-of-sight propagation paths from the reader to several tags may be blocked, causing disturbance to the received signal. This disturbance might incur errors when using the signals for relative localization. Mitigating this impact, however, is challenging in existing RFID technology.
- **Untidy object position**. In practice, some objects are roughly vertical, but not aligned strictly on shelves, like the example shown in the black rectangle in Fig. 1. Our relative localization system should be resilient to such misalignments.

The contribution of HMRL is summarized as follows.

- We design HMRL, a light-weight relative localization system to extract the two-dimensional sequences of a tag array based on the RSS values and phases of backscatter signals. Different from prior work, HMRL utilizes human movements, instead of any device movement, and provides more flexibility and scalability to modern inventory and logistics applications.
- HMRL is resilient to the device diversity and multiple-person movements. This merit enables HMRL to operate and scale well in real deployments.
- We implement a prototype of HMRL using commodity off-the-shelf (COTS) devices and conduct extensive experiments to evaluate its performance. The results demonstrate that HMRL is an efficient and accurate system for tag relative localization with static devices.

The rest of paper is organized as follows. We review related work in Section II. The problem specification of this paper is presented in Section III. In Section IV and Section V, we introduce our system design and analysis. The experiments and evaluation is illustrated in Section VI. Finally, we conclude this paper in Section VII.

II. Related Work

Using RFID for Localization is an attractive research direction. Prior work falls into two categories: absolute localization and relative localization.

Absolute localization: Researchers found that the information contained in the RF signal can be leveraged to infer the locations of RFID tags. LANDMARC [6] is one of the pioneering work of fingerprinting based localization for RFID tags. It utilizes the RSSI similarity between the target tag and reference tags for localization. To setup the fingerprints for all possible locations, fingerprint based approaches [10][14] usually rely on densely pre-deploying tags in the area of interests and pre-collecting their RF signals. Meanwhile, there is a growing interest in using phase differences [11][5][13] or Angel of Arrival (AOA) [3] to estimate the absolute locations of tags. Liu *et al.* [5] utilize the phase differences received at different antennas to localize tags which are on the same horizontal level with those antennas. Their method has been widely adopted by many following solutions [11]. It is also common to utilize AoA information for localization [3]. In particular, some researchers employ advanced electromagnetism and communication techniques, such as the synthetic aperture radar [10] and multiple antennas [13],to achieve accurate localization.

Relative localization: Relative localization is increasingly important for emerging applications of RFID systems. The methods proposed in [7] and [8] are two prominent ones. OTrack [7] leverages a phenomenon that when a tag is close to the reader, the reader can experience higher response reception ratio from the tag. The work proposed in [8] utilizes the spatial-temporal phase profiling to localize objects in two-dimensional space with a moving reader. By analyzing the phase profiles of tags, the system can obtain the spatial order of tags. The major limit of existing RFID based relative localization is the requirement of continuing movement of readers or tags to get necessary signal changes.

Different from previous work, our approach (HMRL) makes use of signal changes caused by the movement of human beings who carry no device, which significantly enlarges the application domain and cost-efficiency of relative localization. Besides object localization, researchers have worked on many other domains by utilizing different types of RF signals, such as [9][4][12][1]. Limited by the space, we do not introduce them in detail.

III. Problem Specification

In this section, we specify the problem that the paper focuses on.

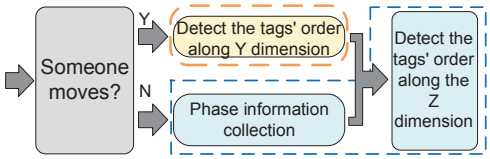


Fig. 2. System work flow

We assume our system is aiming to determine the relative locations of tags that are deployed in a library, warehouse, or supermarket. Each object is pasted by a passive RFID tag. These tags have different IDs (denoted as EPCs according to the EPC C1G2 Global Protocol [2]). All objects of interests are placed in sequence on layered shelves. The tags in a shelf *roughly* form a grid. In each row of this grid, *i.e.*, each layer of the shelf, the tags are horizontal. But in each column of the grid, the tags are roughly vertical to each other, but not strictly aligned. A RFID reader is used for interrogating the tags. Our goal is to identify the vertical and horizontal orders of these tags in the two-dimensional plane on the shelf.

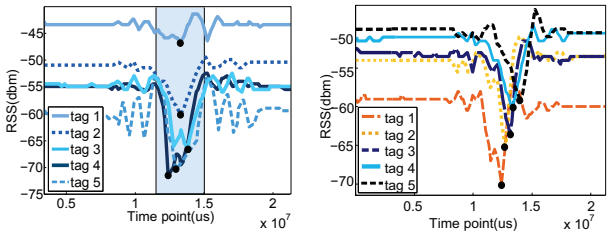
To achieve this goal, we model the tag group as a two-dimensional array. For simplicity, we only focus on the foremost tags in the shelf, as shown in Fig. 1. We setup a three-dimensional coordinate system, with its origin at the center of two reader antennas. We focus on the two-dimensional plane that contains the tag array. In this plane, the tags with a same y coordinate or z coordinate are in a column or row, respectively. We assume there is one or more persons walking in front of the shelf arbitrary, like customers in a shopping mall or employees. By scanning the tags attached to the objects, we can obtain their IDs and signal information, including the Doppler shift, phase, Received Signal Strength(RSS), etc. To obtain the order that the tags are along the Y dimension, we try to investigate the relationship between human movement and the RF changes of tags. We utilize the hyperbolic positioning to sort tags along Z dimension. We will show above techniques in Section. IV. Note that this model is no simpler than existing relative localization methods [7][8].

IV. System Design

In this section, we elaborate the design of HMRL in details. As aforementioned in Section III, the tags attached to the objects on the shelf actually form a matrix. The matrix is on a two-dimensional space, as illustrated in Fig. 1. Relative localization of the tags in this two-dimensional space can be addressed by identifying their order along the Y dimension and Z dimension, respectively.

A. System overview

Fig. 2 outlines the work flow of HMRL. First, we determine whether there is one or more person moving inside the area of interests. If YES, we detect the tags' order along the Y dimension by analyzing the RSS changes caused by the moving person on these tags. In particular, we propose a new metric, namely 'influenced region', to describe such changes. Utilizing this metric, we can cluster the tags with a roughly similar z coordinate, *i.e.*, the tags are placed in a same column in the matrix. Note that if there is no person moving in the area, we can collect the phase information for localization at a later time. Second, we determine the vertical order of tags in the clustered columns. For each column, we detect the order of tags along the Z dimension. In this way, we can identify the relative locations among the tags in a same column. To do so, we utilize a hyperbolic method by measuring unperturbed phase differences received at two antennas. This step uses two kinds of information, the signal phase information of tags when there is no moving person and the clustering results along the Y dimension. For every time that a person moves in front of the shelf, HMRL will report the tag's two-dimensional orders.



(a) tags in a same column (b) tags in different columns

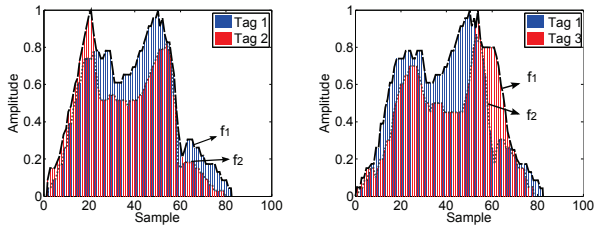
Fig. 3. Comparison of tags in a same and different columns

B. Order detection along the Y dimension

We try to first determine the order of tags along the Y dimension. We resolve the problem of detecting the order along the Y dimension into two problems, namely *'which tags are in a same column'* and *'which column these tags belong to'*. The solution for the former one will classify all tags into several groups. Solving the latter problem then deterministically maps each of these groups to a column in the tag array.

1) Which tags are (roughly) in a same column: We aim to leverage the RSSs change influenced by human movement in this problem. We define the signal segment under this influence as a *influenced region*. In order to thoroughly understand the influenced RSS, we conduct experiments to investigate the appearance of influenced regions in the RSS curve. We ask a volunteer to walk from one side to the other side of the shelf. The layout of the tag array on this shelf is a 5×5 tag grid. Each cell in the grid is 10 cm $\times$ 20 cm ($L \times H$). We depict the results of 5 tags in Fig. 3. We find that such a region is like a bathtub. At the start point of this region, there is a sudden decrease in RSS. The influence may take its effect for a short time period, and then end up with a sudden increase in the RSS curve. Intuitively, we have several parameters about the influenced region potentially used for the clustering, including the start/end time point, the lowest RSS point and the duration of the influenced region. Fig. 3(a) shows the RSSs of those tags when they are in a same column and Fig. 3(b) shows their RSSs when they are placed in different columns. From the result, we find that all the parameters mentioned above are distributed arbitrarily no matter what disposition of these tags is. So it is infeasible to utilize these aforementioned parameters for clustering the tags. Therefore, we have to pursue another methodology.

We propose a new metric, called Overlapped Area of Influenced Region (OAIR). It is also derived from our ob-



(a) $similarity = 0.9739$ (b) $similarity = 0.6878$

Fig. 4. Similarity of a pair of tags in a same and different columns

servation on a comprehensible phenomenon. We find that if a group of tags are roughly in a same column, they almost experience the influence from the moving person *at the same time*, considering the person's moving direction is along the Y dimension. On the contrary, if the tags are in different columns, their influenced regions start at different time points and experience difference time durations. Meanwhile, the influenced regions are very likely overlapped for the tags with close distances. In particular, the tags in different columns, though their start (or end) time points may be very close to each other, their overlapped regions will be shrunken if increasing the interval distance between their columns. This observation inspires us to use the overlapped area to represent the similarity between two tags' influenced regions, and hence to judge whether they are in a same column. For a fair comparison, we use the absolute values of signals and normalize them to $[0, 1]$. Specifically, we calculate the proportion of the overlapped area to the union of the two tags' influenced regions as the similarity, *i.e.*,

$$similarity = \frac{Overlapped\ areas}{Union\ of\ influenced\ regions} \quad (1)$$

Clearly, this $similarity$ is suitable for comparing the overlapped influenced regions between two tags. We thereby define the OAIR of a pair of tags as their similarity.

To demonstrate the feasibility of OAIR, we select two tags from the array and observe their influenced regions. As shown in Fig. 4, f_1 (the black dash line) is the boundary of union influenced regions of tag 1 and 2, while f_2 (the gray dotted line) is the boundary of overlapped areas. Fig. 4(a) and 4(b) show the RSSs overlapped region of tag pair in the same and different columns, respectivley. The OAIR of tags in the same column is much higher than the value of tags in the different columns. It is worth to note that though the start and end time points of the two tags are very close to each other, OAIR can still distinguish them in the latter case. In addition, OAIR utilizes the similarity between influenced regions, which is resilient to the vertical misalignments of the objects in different layers on the shelf. In the following, we define the *tolerant distance*, denoted as d_t, to quantify the distance between misaligned tags in different layers.

To cluster all tags in the array, we calculate the OAIR of each pair of tags and put the result in a matrix $A_{n\times n}$, where n equals to the total number of tags in the array. We still use the 5×5 tag array for illustration. For simplicity, we only choose 6 tags, whose layout is actually a sub-matrix with 3 rows and 2 columns. Specifically, tag 1, 3, 5 are in the first column and tag 2, 4, 6 are in the second column. The volunteer walks through the first column and second column sequentially. Table I demonstrates the OAIR results of the six selected tags in their $A_{6\times 6}$ array. For each tag in the matrix, our goal is to determine the top M tags that have the highest OAIR, which are most likely in a same column with this tag. In practice, the number of objects in one column can be known in advance. For instance, it is very common that this number is usually equal to the number of layers in the shelf. So we can set M as the number of layers in the shelf.

TABLE I
OAIR RESULTS BETWEEN ALL TAG PAIRS

No. \ No.	1	2	3	4	5	6
1	1	0.9706	0.9888	0.8759	0.9951	0.9380
2	0.9706	1	0.9079	0.9553	0.9684	0.9918
3	0.9888	0.9079	1	0.7972	0.9875	0.9041
4	0.8759	0.9553	0.7972	1	0.8667	0.9909
5	0.9951	0.9684	0.9875	0.8667	1	0.9506
6	0.9380	0.9918	0.9041	0.9909	0.9506	1

index \ top	1	2	3
1	1	5	3
2	2	6	1
3	3	1	5
4	4	6	2
5	5	1	3
6	6	2	4

Fig. 5. Table T

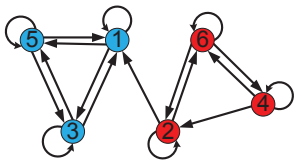


Fig. 6. Digraph for reflecting the tag's top M list

After choosing the top M tags, we obtain a new $n \times M$ table $T_{n\times M}$ for all n tags. Each element (i, t) in the table $T_{n\times M}$ represents the t-th tag in the tag i's top M list. For example, the element $(4, 2)$ in Fig. 5 is 6, which indicates that tag 6 has the second highest similarity with tag 4 in its top M list. Then we transfer table T into a digraph, in which each edge is correlated an element in T. For example, Fig. 5 can be mapped to a digraph, as plotted in Fig. 6.

From this digraph, we have two observations: *1)* any pair of tags in a same column have a pair of edges with each other; *2)* two tags in different columns only have a single edge or no edge at all. From the digraph Fig. 6, if we remove all single edges among the vertices, we can get several independent subgraphs. One is formed by tag 1, 3, and 5, while another contains tag 2, 4, and 6.

To use the above idea, we first transform table T into an adjacent matrix B. The value of component b_{ij} in matrix B is defined as follows:

$$b_{ij} = \begin{cases} 1 & j \in \Phi_i \\ 0 & j \notin \Phi_i \end{cases} \quad \Phi_i = \{\forall j | \exists < i, j >\}, \quad (2)$$

where Φ_i is the set of all vertices in the digraph that have an direct edge $< i, j >$ between the pair of vertices, i and j. In other words, set Φ_i contains the top M candidate tags that are in a same column with i. For example, the adjacent matrix B generated from the digraph in Fig. 6 is shown in Eq. 3.

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 3 & 1 & 3 & 0 & 3 & 0 \\ 1 & 3 & 1 & 2 & 1 & 2 \\ 3 & 1 & 3 & 0 & 3 & 0 \\ 0 & 2 & 0 & 3 & 0 & 3 \\ 3 & 1 & 3 & 0 & 3 & 0 \\ 0 & 2 & 0 & 3 & 0 & 3 \end{bmatrix} \tag{3}$$

Based on matrix B, we remove the single edges and identify the column for each tag. We calculate $R = B \times B^T$. Here B^T is the transpose of B. Obviously, only when the arrows $< i, j >$ and $< j, i >$ both exist in the digraph, *i.e.*, the b_{ij} in matrix B and b'_{ij} in matrix B^T are all equal to 1, the product will be equal to 1. Otherwise, the product will be 0. In this way, we remove the single edge in the digraph. The matrix R corresponding to the graph in Fig. 6 is shown in Eq. 3.

Each component r_{pq} in matrix R reflects the number of common members in the Φ_p and Φ_q, *i.e.*, the number of common candidate tags in both the top M lists of tags p and q. That is:

$$r_{pq} = |\Phi_p \bigcap \Phi_q| \tag{4}$$

Here we use the value of r_{pq} as a weight for determining whether tags p and q are in a same column. Specifically, the larger the weight is, the higher probability that the tags p and q are in a same column. In practice, for any pair of tags p and q, the maximum number of their r_{pq} is limited to M. Therefore, we multiple matrix R by $1/M$ and compare the results with a threshold $thre$. This threshold can be empirically determined, as described in Section VI. In this way, we cluster the tags into different groups, and in each group the tags are in a same column.

2) To which column these tags belong: After clustering the tags, we need to determine the relative order of these clusters, *i.e.*, to which column these clusters belong. A straightforward way is to place a number of reference tags in a row on the shelves. The position of each reference tag is known in advance for localizing any tags nearby. However, such a scheme is impractical, since for any tag in a given cluster the system has to search all reference tags to determine which one is nearest the tag.

Instead of using this treatment, we paste two reference tags aligned to the rightmost and leftmost columns of a shelve. It is obvious that in the shelf each column is adjacent to no more than two columns. Note that the rightmost and leftmost columns are only adjacent to one column. According to this observation, we propose Algorithm 1. In this algorithm, we first assign a serial number s_t to each column according to the time sequence that it is influenced. For each column s_t, we find out two columns s_t^1 and s_t^2, which are with the top two weights in *weight(t,:)*. Obviously, s_t^1 and s_t^2 are the two adjacent columns of s_t. After identifying these two columns, the next task is to determine their positions relative to s_t, *i.e.*, on the left-hand or right-hand side of column s_t. Suppose their relative positions are s_t^1 and s_t^2. Above process is designed as an adjacent function $F(s_t)$, where $F(s_t) = \{s_t^1, s_t^2\}$. On the other hand, we can easily find out the rightmost and leftmost

Algorithm 1 Determining the order of columns

Initialization:

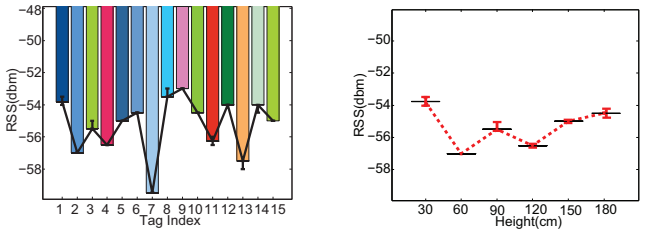
The influenced column sequence $S = s_1 \rightarrow s_2 ... \rightarrow s_t ... \rightarrow s_T$, $(T \geq N)$ (N is the number of columns);

The rightmost and leftmost reference tags ref_r and ref_l.

Output:

The columns' order from left to right is $C = c_1, c_2, ...c_N$;

1: $p = 1$, $q = 2$;
2: **while** $q \leq T$ **do**
3: **if** $\exists s_p \rightarrow s_q$ **then**
4: $weight(p, q)++$, $weight(q, p)++$; $p++$, $q++$;
5: **end if**
6: **end while**
7: Define an adjacent function $F(s_t)$ as $F(s_t) = \{s_t^1, s_t^2\}$;
8: Find out the rightmost and leftmost columns c_1 and c_N.
9: Calculate the column order C by linking $F(S)$.
10: **return** C;



(a) Different tags at a same position (b) A tag with different heights

Fig. 10. Tag diversity and multi-path effect

columns s_{left} and s_{right} by comparing the influenced area of all columns with reference tags ref_l and ref_r. Assume that the the leftmost and rightmost columns are c_1 and c_N. Then we have $c_1 = s_{left}$ and $c_N = s_{right}$. As aforementioned, c_1 and c_N are only adjacent to one column *i.e.*, $c_2 = s_{left}^1$ and $c_{N-1} = s_{right}^1$. Combining the relative positions between each column and its adjacent columns, we can determine the relative sequence one by one either from left to right or vice versa. For example, since we have known the index of c_1, we can determine c_2, using the information of relative positions between c_1 and c_2. Recall that two adjacent columns of c_2 have been identified, *i.e.*, $F(c_2) = \{c_1, c_3\}$. Hence the relative position of c_3 can be also determined. Iteratively, we can obtain the columns' order C.

C. Detect tags order along Z dimension

Compare the Y dimension's ordering, vertically determining the relative locations for tags in a same column is even more difficult.

Due to the multi-path effect, tags with different heights might not show distinct differences on their received signal strength(RSS). As shown in Fig. 10(b), the RSSes from a given tag do not vary linearly when the distance increases. In addition, the hardware diversity among tags is non-trivial. Fig. 10(a) shows the RSSs of different tags at a same position, which indicates the influence of hardware diversity. As a result, the RSS of a given tag cannot be deterministically mapped to the distance from the tag to the reader.

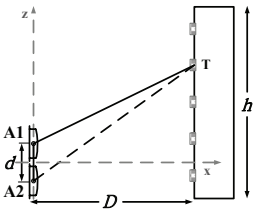


Fig. 7. System deployment

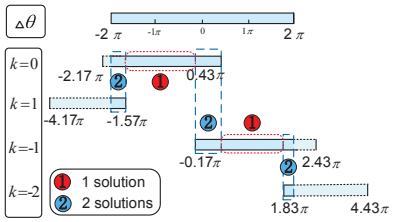


Fig. 8. Possible solutions of k

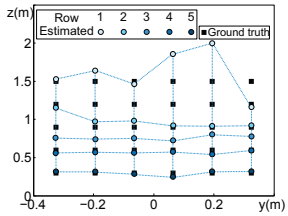


Fig. 9. 2D relative localization for the tag array

Therefore, we have to pursue a method that does not rely on the tags' RSS for relative localization. Our basic idea is constructing a number of spatial hyperbolas by measuring the phase differences received at two near antennas. There should be some intersections of those hyperbolas and the two-dimensional plane that contains the tag array. These interactions are potential locations of the tags. In this way, we are able to determine the tags' order by identifying the most possible intersections.

We illustrate the above idea in Fig. 7. We employ two reader antennas. One is on the ground. Another is deployed vertically above it. Both of them face towards the tag array. We establish a 3D coordinate system, in which the origin is at the center of demarcation line of two antennas. The x-axis is perpendicular to the shelves and z-axis is vertical to the ground, as shown in Fig. 7. Note that the y-axis is paralleled to the shelf, which is not shown in Fig. 7. We use A_1, A_2, and T to denote the centers of antenna 1, 2 and tag, respectively. The distance between A_1 and A_2 is denoted as d. We adopt two mainstream directive antennas whose size is 25 cm$\times$ 25 cm. So $d = 25$ cm. The distance between A_1 (or A_2) and T is shown in Eq. 5.

$$\begin{aligned} d(A_1,T) &= \frac{\frac{1}{2}\cdot(2k_1\pi+\theta_1+\theta_{t_i})}{2\pi}\cdot\lambda \\ d(A_2,T) &= \frac{\frac{1}{2}\cdot(2k_2\pi+\theta_2+\theta_{t_i})}{2\pi}\cdot\lambda, \end{aligned} \tag{5}$$

where θ_1 and θ_2 are the received phase values of Tag_i at antenna 1 and 2, respectively. θ_{t_i} is the initial rotation of Tag_i. Note that there is a "round-trip" for the RF signals received at the antenna. Hence we multiply the length path by $\frac{1}{2}$ in Eq. 5. Since the phase is periodic, k_1 and k_2 are two nonnegative integers whose values can be $0, 1, 2, ..., K, (K \to \infty)$.

Locating a given tag requires to determine the value of k_1 and k_2, which implies that we should alleviate the influence of individual differences of tags. We then subtract $d(A_2,T)$ from $d(A_1,T)$ in Eq. 5 and find that the initial phase θ_{t_i} can be eliminated. According to the triangle inequality, *i.e.*, the difference (absolute value) between any two sides of a triangle should be less than the length of the third edge. Thus, the absolute value of the difference between $d(A_1,T)$ and $d(A_2,T)$ should be less than the distance between A_1 and A_2. Hence we have:

$$\frac{\lambda}{4\pi} \mid 2k\pi + \triangle\theta \mid < d, \tag{6}$$

where $k = k_1 - k_2$ and $\triangle\theta = \theta_1 - \theta_2$. Note that each possible k corresponds to a hyperbola, *i.e.*, a possible solution of tag's position T. In practice, there should be only one k corresponding to the real location of tag T. However, k has 5 possible candidates according to Eq. 6, including $k = 0, \pm1, \pm2$ (here the detail is omitted because of the space limit). We need to identify this feasible k.

Fortunately, in practice the size of shelves facilitates identifying this k. We assume that the distance between the antennas and shelf is D and the height of shelf is h. These two parameters are usually constrained in real implementations. For example, the shelf used in shopping malls or libraries are usually designed for people to take items conveniently. Thus, the shelf height would not exceed the average human height too much. Similarly, the interval distance between two adjacent shelves should be sufficiently long for enabling multiple costumers' passing. Under this circumstance, the value ranges of D and h can be determined via such prior-knowledge, which shrinks the value range of k.

To illustrate how HMRL discards the infeasible ks, we take an example of $D = 1.8$m and $h = 2$m, which are two general settings in real-world. For simplicity, we assume tag T's $y = 0$. For the tags in the column, their coordinates range from (1.8,0,-0.25) to (1.8,0,1.75). Therefore, we are aiming to determine the z coordinate of tag T within this range. We can obtain the value of $d(A_1,T) - d(A_2,T)$ according to the Pythagorean theorem (the calculation is omitted here because of the space limit). According to Eq. 6, we have:

$$\frac{4\pi}{\lambda}\cdot min \le (2k\pi + \triangle\theta) \le \frac{4\pi}{\lambda}\cdot max, \tag{7}$$

where min and max are the minimum and maximum value of $d(A_1,T) - d(A_2,T)$, respectively. In above example, the range of $d(A_1,T) - d(A_2,T)$ is $[-0.1742, 0.0343]$. To obtain the possible value of k, we exhibit the possible range of $\triangle\theta$ and its corresponding values of k in Fig. 8. Because both θ_1 and θ_2 are received phases of tag T, their difference $\triangle\theta$ is in the range of $(-2\pi, 2\pi)$. We find that for some observations of $\triangle\theta$, only one k conforms all the conditions above. We mark it with a circle 1 in Fig. 8. And we mark the phase difference range that holds two possible values of k with circle 2. As shown in Fig. 8, when the value of $\Delta\theta$ is within $(-1.57\pi, -0.17\pi) \bigcup (0.43\pi, 1.83\pi)$, the number of k is only 1. On the other hand, when the value of $\Delta\theta$ is out of this range, the number of possible k increases to 2. In this way, we can shrink the number of k to 2. Given each k, we have a corresponding height of tag T.

Since there may be two possible intersections, it is still unclear what order the tags are in a same column, because each possible k may introduce a solution of the order. Next, we apply Linear Programming to ultimately find out the unique solution fork. Adopting LP in HMRL is based on two

constrains: a) some tags have only one possible solution. We named these tags as *anchor tags*. Other tags that are more than one possible solutions are defined as *undetermined tags*. Obviously, each anchor tag has a deterministic z coordinate; b) The vertical distance between two adjacent layers in the shelf is known in advance. Suppose this distance is ℓ. In this example, we assume that ℓ is identical in the shelf, *i.e.*, $z_{upper} - z_{lower} = l$ (where z_{upper} and z_{lower} are the z ordinates of two adjacent layers). Note that in practice l is adjustable. Combined with these two constrains, we can determine the relative order for tags by minimizing the residual errors as follows:

$$min_{\hat{h(i)}}\{\sum \parallel \hat{h(i)} - m_i \cdot \ell \parallel^2 + \sum \parallel h(j) - m_j \cdot \ell \parallel^2\} \tag{8}$$

where $h(j)$ represents the z ordinate of anchor tag j and $\hat{h(i)}$ represents the z ordinate of undetermined tag i. Here m_i and m_j are the relative order of i and j, respectively. For each undetermined tag i, there is an intersection of hyperbola and shelves given a possible value of k. However, the height of this intersection $\hat{h(i)}$ may yield different tags' order m_i. The intuition behind Eq. 8 is that we pursue all possible ks for all undetermined tags and find out the one with the minimum residual error, which is correlated to the correct order.

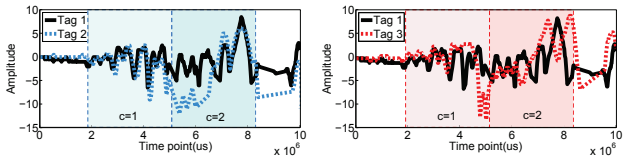
For example, if an undetermined tag has two possible intersections, *i.e.*, $\hat{h(i)} = 0.1$ and $\hat{h(i)} = 1.1$. Given that the heights of the anchor tags are $h(j_1) = 0.6$ and $h(j_2) = 0.3$, there are two estimated results for the order m_i of the undetermined tag. Since the $\hat{h(i)} = 0.1$, $m_i = 1$, indicating that the tag is on the lowest layer. While for $\hat{h(i)} = 1.1$, $m_i = 3$, suggesting that the tag is on the highest layer. Accordingly, the optional orders of the anchor tags m_{j_1} and m_{j_2} are $2, 3$ and $1, 2$. If $\ell = 0.3m$, the residual errors are $[(0.1 - 0.3)^2] + [(0.3 - 0.6)^2 + (0.6 - 0.9)^2] = 0.22$ when $\hat{h(i)} = 0.1$ and $[(1.1-0.9)^2]+[(0.3-0.3)^2+(0.6-0.6)^2] = 0.04$ when $\hat{h(i)} = 1.1$. Obviously, the latter is the correct order.

The effectiveness of above solution for the vertical relative localization is demonstrated in Fig. 9. In this figure, the black squares are the actual positions of tags and the blue circles are the estimated results using our hyperbola based method. We can see that our method is reliable in retrieving the tags' order along the Z dimension.

V. Scenario of Multiple Moving Persons

It is very common that multiple persons are moving in the deployment area of HMRL at a same time. Since HMRL does not rely on the human movement to determine the Z dimension order, we only need to detect the Y dimension order under the scenario of multi-persons.

Different from the case of single-person, multiple moving persons may influence multiple columns of tags simultaneously, raising the difficulty of relative localization along the Y dimension. Under this circumstance, the influence from multiple moving persons is much more arbitrary in both the spatial and temporal dimensions. In addition, we have no



(a) Signals of tags in a same column (b) Signals of tags in 2 columns

Fig. 11. Signals in the multi-person case

idea about the real positions of those moving persons, not to mention their influences to the tags.

Fortunately, our OAIR method can still function well in the multi-person case. The basic idea of OAIR is that the tags in a same column will always be influenced at the same time, no matter how many persons move horizontally. While the tags in different columns do not follow this principle. So we can just compare the signals between each pair of tags within a same time duration w, which denotes the length of a time window c ($1 \leq c \leq C$). If the two tags are in a same column, their OAIR will always be much higher than those pairs of tags that are in two different columns. To exam the effectiveness of OAIR, we observe the RSSs of 3 tags in our tag array when 2 volunteers walk simultaneously. Tags 1 and 2 are in a same column while tags 1 and 3 are in two adjacent columns. The horizontal distance between tags 1 and 3 are 10 cm. We present the variation tendency in Fig. 11. In the first time window $c = 1$, the RSS changes of tags 1, 2, 3 are indistinguishable. However, in the time window $c = 2$, tag 1 and 3 show significant difference in their RSSs.

To realize above comparison, we try to combine estimation results in several time windows. In each window c, we redefine the OAIR similarity as follows:

$$similarity = \frac{\int_{c_s}^{c_e} f_2 dx}{\int_{c_s}^{c_e} f_1 dx}, \tag{9}$$

where f_1 and f_2 are the union of influenced regions and overlapped area of tag paris, respectively. c_s and c_e are the start time stamp and end time stamp of window c. We record the weight matrix R_c with newly generated OAIRs. At the end of this time period, we multiply these R_cs successively as following.

$$R' = \frac{\prod_{c=1}^{C} R_c}{M^C}, \tag{10}$$

where R' is a weight matrix by accumulatively combining multiple estimation results. After such multiplication, the correct result in R_cs will be amplified and strengthened, while the incorrect ones may be suppressed. Note that we can also utilize Algorithm 1 to infer the order of columns under the multi-person situation.

We are further motivated to combine multiple estimation results in the single person case for improving the accuracy. As aforementioned, every time a person passing the shelves offers an opportunity for a relative location estimation. To obtain a more accurate classification result, we try to combine multiple estimations derived from multiple passings of single-person. We screen out the signal segments that influenced by multiple single-person-passings, and divide these signal segments into several time windows. The estimated matrix of the c^{th} time window is denoted as R'_c. Similar to the

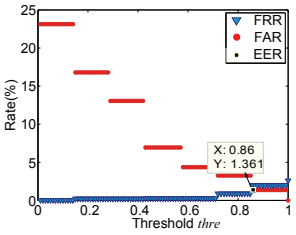


Fig. 12. Threshold selection

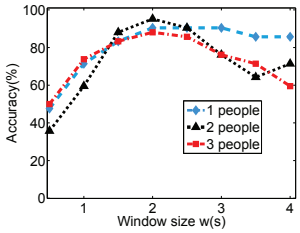


Fig. 13. Window size selection

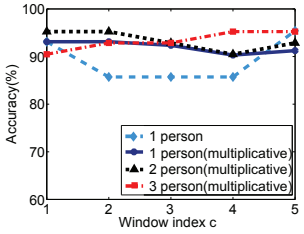


Fig. 14. Accuracy of OAIR

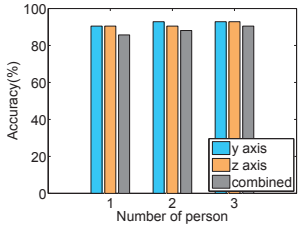


Fig. 15. Localization accuracy when c=5 and w=2s

multi-person case, we multiply these R'_cs like Eq. 10. We evaluate the effectiveness of this combination in Section VI.

VI. Experiment and Evaluation

A. Implementation

Hardware: We implement an HMRL prototype, which consists of a COTS UHF RFID reader model Impinj R420, two directional antennas in model IPJ-A0311, and a set of passive tags. The tags include five different types: Alien-964X, Impinj E41B, E41C, H47, and AZ-E53. Note those 5 types of tags are in different sizes and shapes.

System setup: To be consistent with the real implementation, the horizontal distance between the reader antennas and the shelves is 1.2 m $\sim$ 2.4 m. We utilize a tag array that is with 7 rows and 6 columns, as well as 2 reference tags. In our experiments, we invite 5 volunteers, varying from 1.6 m to 1.8 m in their height and from 46 kg to 77 kg in their weight. The average walking speed of these volunteers is about 1.5 m/s . To account for the hardware diversity, we mix all types of passive tags together in the tag array. The height of each layer ranges from 20 cm to 40 cm. While the distance between two adjacent tags in a same row varies from 5 cm to 40 cm.

Metrics: We mainly use the accuracy to quantify the HMRL's performance. The metric *Accuracy Ratio* is defined as follows:

$$Accuracy = \frac{\#\ of\ tags\ ordered\ correctly}{\#\ of\ tags\ in\ total}. \quad (11)$$

We define that a tag is ordered incorrectly in a sequence of tags if its detected order is not equal to its actual location. For example, if a tag sequence 'ABCDE' is wrongly recognized as 'ABCED', the accuracy will be 60%. We also utilize False Reject Rate (FRR) and False Accept Rate (FAR) to further evaluate the accuracy. Note we measure the accuracy based on only one report. In actual applications, the system can output the locations for every person walking through and the accuracy can be improved by considering multiple times of localization.

B. Parameter Selection

We try to determine two important parameters for HMRL in real-world applications, *i.e.*, the threshold $thre$ of weight matrix R and time window's size w.

Threshold $thre$: As aforementioned in Section. IV, the threshold $thre$ plays an important role in judging whether two tags p and q are in a same column. In general, the higher the threshold $thre$ is, the more efficient the top M tags will be identified. However, on the other hand, the lower the $thre$ is, the more inaccurate the system will be. It is necessary to balance two factors while determining a proper $thre$. We utilize FAR and FRR to represent the performance of these two factors. In Fig. 12, we depict the Equal Error Rate (EER) of HMRL. EER is a common metric to evaluate the accuracy performance for recognition systems. If we choose the threshold at the EER point, the sum of FAR and FRR reaches its minimum, and these two metrics are optimally balanced. With this EER point, we can properly determine the threshold. The default $thre$ is set as 0.86 in our prototype HMRL. In practice, $thre$ can be experimentally adjusted according to users' demands.

Window size w: As aforementioned in Section. V, we perform a matrix multiplication on the R_cs at the time window c successively. In general, a proper window size w results in high accuracy. As shown in Fig. 13, we exhibit the accuracy of HMRL with $1 \sim 3$ persons moving inside the area, respectively. The x axis represents the window size w, which varies from 0.5s to 4s. We find that when $w = 2$s, The accuracy of HMRL is high, *i.e.*, about 90%, for all these three cases. With this window size, HMRL is also accurate in the single-person case. So we employ $w = 2$s as the setting of time window size.

C. Evaluation

After determining critical system parameters, we evaluate the performance of HMRL in both the single-person and multi-persons cases. In addition, we discuss the effect of combining multiple estimation results.

We depict the HMRL's accuracy when performing relative localization along the Y dimension in Fig. 14. The x axis is the window index. For the single-person case, we exhibit the accuracy before and after combining multiple estimation results. And for the multi-person case, we only focus on the result after the multiple measurement. The result shows that combining multiple measurement results helps HMRL to maintain a high accuracy (about $90.24\% \sim 93.11\%$). In addition, in the multi-person case HMRL shows a slightly higher accuracy (about $90.48\% \sim 95.24\%$) than in the single-person case.

We also show the results that perform relative localizations along the Y dimension, Z dimension, and in the entire two-dimension space in Fig. 15. We find that the relative localization accuracy along the Y and Z dimension is 91.24% and 90.48%, respectively. These results are higher than that (90%) in single-person case. And for the relative localization in the entire 2D space, the accuracy is 88.71%, 90.48%, 91.24% when there is/are 1, 2, and 3 persons, respectively.

To detect the tolerant distance d_t, we also conduct an experiment in which we vary the average misaligned distance between the tags in a same column from 0.5 cm$\sim$ 5 cm. The

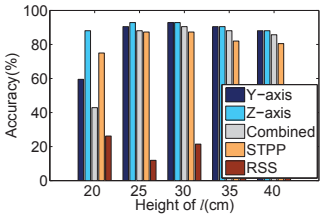


Fig. 16. Layer height

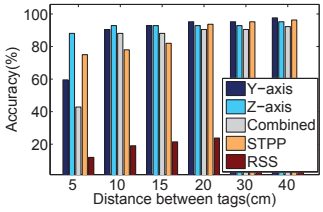


Fig. 17. Interval distance

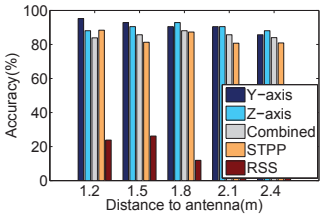


Fig. 18. The distance from the reader to shelves

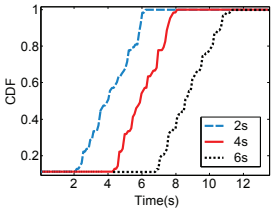


Fig. 19. CDF curves if somebody blocks tags for a while

TABLE II
ACCURACY ALONG THE Y DIMENSION WHEN VARYING d_t

d_t(cm)	0.5	1	2	3	4	5
Accuracy	0.881	0.881	0.857	0.809	0.762	0.524

distance between two adjacent columns is 10 cm. The results are listed in Table. II. We find that the accuracy is higher than 0.809 even if $d_t \leq 3$ cm, demonstrating that HMRL is resilient to misalignments.

In a short summary, HMRL can achieve accurate two-dimensional relative localization in practice. Besides, combining multiple estimation results indeed improves the accuracy of HMRL.

We then investigate the performance of HMRL by tuning the prototype system's settings, including the layer height, interval distance between adjacent tags and distance between the shelf and reader. We show the result of using HMRL under these tunings along the Y dimension, along the Z dimension, and in the entire 2D plane. We also compare HMRL with the approach utilizing RSS directly and a state-of-the-art relative localization approach, STPP [8].

Layer height vs. Accuracy: We vary the layer height from 20 cm to 40 cm. The accuracy is shown in Fig. 16. We find that except $\ell = 20$ cm, HMRL outperforms both STPP and RSS in terms of the accuracy of 2D relative localization.

Tag separation distance vs. Accuracy: To simulate the real-world layout of objects, we tune the interval distance between two tags in each layer within the range of [5 cm, 40 cm]. The localization results are shown in Fig. 17. We find that the larger the distance is between tags, the higher the accuracy HMRL can achieve along the Y dimension. The 2D relative localization accuracy of HMRL is always higher than 88.10%.

Reader distance to shelves vs. Accuracy: We then adjust the distance between the shelf and the reader antennas from 1.2m to 2.4m. The results are plotted in Fig. 18. We find that with the increase of above distance, HMRL presents a decrease in its localization accuracy along the Y dimension, but an increase in the one along the Z dimension. Under most situations, HMRL performs better than STPP and RSS.

Blocked tags: In our system, we assume that customers are moving. However, they may stop and stand for a little while. Under this situation, our system cannot read the tags blocked by the customer. HMRL addresses this issue by periodically collecting data within each time window and leverage multiple ordering results to reach a final result. We conduct experiments where a stop-and-go customer moves in the region. He first moves inside and stays static for 2s, 4s and 6s, respectively. Then he passes through the inventory area. The CDF curves are shown in Fig. 19 and the results indicate that HMRL can handle the problem if somebody blocks tags for a while.

VII. CONCLUSION

In this paper, we propose HMRL, a relative localization system for passive tags by utilizing human movement. Compared to prior solutions, HMRL is more efficient and convenient to deploy, thus it enables more applications. The experimental results show that HMRL is accurate and practical in the relative localization for real-world applications.

VIII. ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant No. 61325013, 61572396, 61373175, 61402359. Chen Qian is supported by UC Santa Cruz Startup Grant and National Science Foundation grant CNS-1701681. We thank the valuable comments from anonymous reviewers.

REFERENCES

[1] M. Chen, J. Liu, S. Chen, and Q. Xiao. Efficient anonymous category-level joint tag estimation. In *Proceedings of IEEE ICNP*, 2016.

[2] EPCglobal. EPC^{TM} radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz–960 MHz, 2005.

[3] C. Hekimian-Williams, B. Grant, X. Liu, Z. Zhang, and P. Kumar. Accurate localization of RFID tags using phase difference. In *Proceedings of IEEE RFID*, 2010.

[4] M. Li, P. Zhou, Y. Zheng, Z. Li, and G. Shen. Iodetector: A generic service for indoor/outdoor detection. *ACM Transactions on Sensor Networks (TOSN)*, 11(2):28, 2015.

[5] T. Liu, L. Yang, Q. Lin, Y. Guo, and Y. Liu. Anchor-free backscatter positioning for RFID tags with high accuracy. In *Proceedings of IEEE INFOCOM*, 2014.

[6] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. LANDMARC: Indoor location sensing using active RFID. *Wireless Networks*, 10(6):701–710, 2004.

[7] L. Shangguan, Z. Li, Z. Yang, M. Li, and Y. Liu. OTrack: Order tracking for luggage in mobile RFID systems. In *Proceedings of IEEE INFOCOM*, 2013.

[8] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu. Relative localization of RFID tags using spatial-temporal phase profiling. In *Proceedings of USENIX NSDI*, 2015.

[9] J. Shi, R. Zhang, Y. Liu, and Y. Zhang. Prisense: Privacy-preserving data aggregation in people-centric urban sensing systems. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[10] J. Wang and D. Katabi. Dude, where's my card?: RFID positioning that works with multipath and non-line of sight. In *Proceedings of ACM SIGCOMM*, 2013.

[11] J. Wang, D. Vasisht, and D. Katabi. RF-IDraw: Virtual touch screen in the air using RF signals. In *Proceedings of ACM SIGCOMM*, 2014.

[12] K. Wu, H. Li, L. Wang, Y. Yi, Y. Liu, D. Chen, X. Luo, Q. Zhang, and L. M. Ni. hJam: Attachment transmission in WLANs. *IEEE Transactions on Mobile Computing*, 12(12):2334–2345, 2013.

[13] L. Yang, Y. Chen, X. 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*, 2014.

[14] Y. Zhao, Y. Liu, and L. M. Ni. VIRE: Active RFID-based localization using virtual reference elimination. In *Proceedings of IEEE ICPP*, 2007.