

Full-Dimension Relative Positioning for RFID-Enabled Self-Checkout Services

CHUNHUI DUAN, School of Software, Tsinghua University, China and School of Computer Science and Technology, Beijing Institute of Technology, China

JIAJUN LIU, School of Software, Tsinghua University, China

XUAN DING, School of Software, Tsinghua University, China

ZHENHUA LI*, School of Software, Tsinghua University, China

YUNHAO LIU, School of Software, Tsinghua University, China

Self-checkout services in today's retail stores are well received as they set free the labor force of cashiers and shorten conventional checkout lines. However, existing self-checkout options either require customers to scan items one by one, which is troublesome and inefficient, or rely on deployments of massive sensors and cameras together with complex tracking algorithms. On the other hand, RFID-based item-level tagging in retail offers an extraordinary opportunity to enhance current checkout experiences. In this work, we propose Taggo, a lightweight and efficient self-checkout schema utilizing well-deployed RFIDs. Taggo attaches a few anchor tags on the four upper edges of each shopping cart, so as to figure out which cart each item belongs to, through relative positioning among the tagged items and anchor tags without knowing their absolute positions. Specifically, a full-dimension ordering technique is devised to accurately determine the order of tags in each dimension, as well as to address the negative impacts from imperfect measurements in indoor surroundings. Besides, we design a holistic classifying solution based on probabilistic modeling to map each item to the correct cart that carries it. We have implemented Taggo with commercial RFID devices and evaluated it extensively in our lab environment. On average, Taggo achieves 90% ordering accuracy in real-time, eventually producing 95% classifying accuracy.

CCS Concepts: • **Networks** → **Location based services; Mobile networks**; • **Human-centered computing** → *Ubiquitous and mobile computing*.

Additional Key Words and Phrases: RFID, relative positioning, self-checkout

ACM Reference Format:

Chunhui Duan, Jiajun Liu, Xuan Ding, Zhenhua Li, and Yunhao Liu. 2021. Full-Dimension Relative Positioning for RFID-Enabled Self-Checkout Services. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 1, Article 7 (March 2021), 23 pages. <https://doi.org/10.1145/3448094>

*Corresponding author

Authors' addresses: Chunhui Duan, duanch09@gmail.com, School of Software, Tsinghua University, Beijing, China, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China; Jiajun Liu, jj-liu16@mails.tsinghua.edu.cn, School of Software, Tsinghua University, Beijing, China; Xuan Ding, School of Software, Tsinghua University, Beijing, China, dingxuan@tsinghua.edu.cn; Zhenhua Li, School of Software, Tsinghua University, Beijing, China, lizhenhua1983@tsinghua.edu.cn; Yunhao Liu, School of Software, Tsinghua University, Beijing, China, yunhao@tsinghua.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

2474-9567/2021/3-ART7 \$15.00

<https://doi.org/10.1145/3448094>

1 INTRODUCTION

1.1 Background and Motivation

The concept of checkout-free or self-checkout shopping has swept the retail market in the past few years [3, 31]. It opens up a frictionless retail experience where customers no longer have to spend too much time waiting in a long checkout line during peak hours even for a bottle of water. It is also attractive to store owners because the labor costs for cashiering can thus be greatly saved. Furthermore, self-checkout service provides a bit of extra privacy – no staff directly sees what or how much the customer is buying. Such unique advantages draw much attention from researchers and a range of checkout-free options have been explored, from self-scanning items via a smartphone, to more complex solutions that use sensors and cameras to track shoppers' and goods' movements in the store. However, the former requires shoppers to scan the barcode on each item one by one, which is still time-consuming, while the latter relies on the deployment of a great many high-cost devices.

We turn our attention to the Radio Frequency Identification (RFID) technology that has progressed to a remarkable state recently with broad applications in everyday life, such as product monitoring, goods tracking and item identification [20, 34, 40]. As a key enabler of automatic identification technology, RFID offers an appealing alternative when compared against traditional barcodes, given the merits of non-contact communication, reading ability of fast-moving objects and multiple objects, larger data storage capacity, *etc.* Recent years have also witnessed the cost of RFID tags be brought down dramatically from several dollars to one tenth of a dollar. In a number of today's bookstores and supermarkets, items have already been attached with passive RFID tags for identification [5, 6]. We believe or envision that RFID would facilitate existing barcode-based services in the near future.

To enable automatic checkout, there are some existing demonstrations that let consumers move the shopping carts into a specialized RFID gateway [4], or walk the bag of products to buy through a pre-installed channel [6]. Nevertheless, such schema requires customized hardware efforts, and thus is inconvenient to generalize, let alone the considerable costs. Another feasible solution is to localize tags with a fine granularity – if we can obtain all tags' absolute positions precisely, the shopping carts they belong to could then be inferred. Unfortunately, accurate tag localization in itself is a challenging task, and state-of-the-art methods [33, 34, 40] require dedicated devices (like USRP) or relatively large deployment costs (for reference tags and antennas) in order to achieve high precision.

1.2 Our Schema

In this work, we introduce Taggo, a *lightweight* and *efficient* self-checkout schema. Different from the above-mentioned attempts, we try to offer a more generic option with purely commercial off-the-shelf (COTS) devices. As shown in Fig. 1, a conveyor belt is deployed in the checkout region, moving at a uniform speed to propel the shopping carts of users. Items inside the carts are the products consumers would like to buy, while those outside on the belt are unwanted ones dropped by consumers. In terms of deployment cost, Taggo only adopts two properly arranged RFID antennas connecting to the same reader. In terms of working efficiency, Taggo wants to allow multiple users to check out simultaneously. In other words, there could be several shopping carts on the belt at the same time, placed in either a back-to-back or a side-by-side manner. Then, it is crucial to determine which cart an item belongs to.

Taggo resolves the checkout problem through *relative positioning* among RFID tags, which is an emerging technique put forward in recent years [22, 30]. Concretely, we attach a few *anchor tags* on the front, back, left, and right edges of each shopping cart. To determine whether one tagged object belongs to a specific cart, Taggo tries to get the *relative location* (or says the order) of this tagged object with respect to the anchor tags without knowing their absolute positions.

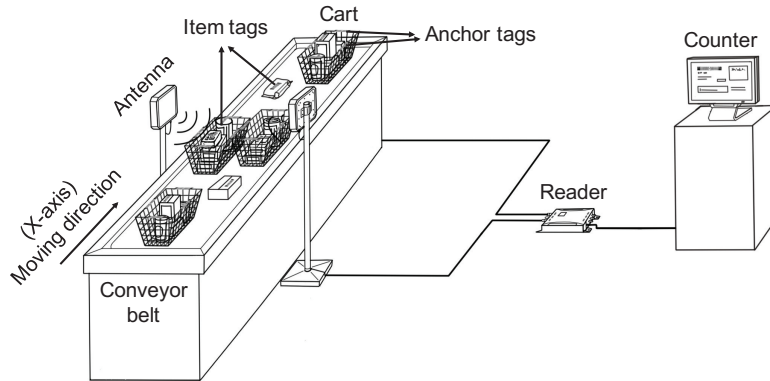


Fig. 1. Architectural overview of Taggo. Shopping carts are propelled by a conveyor belt moving at a uniform speed. We establish the coordinate system regarding the plane of the conveyor with the X-axis parallel to the moving direction. Each item is attached with a passive RFID tag, and anchor tags are deployed on the four surrounding (*i.e.* front, back, left, and right) edges of each cart.

1.3 Challenges and Solutions

Putting this idea into practice, the challenges we face are mainly the following folds. *First*, state-of-the-art studies in the relative positioning domain [22, 30] mainly focus on the ordering of tags in one dimension, which apparently cannot meet the needs in our checkout scene. As a user-friendly schema, Taggo should not have restrictions on the displacement of users' shopping carts. In other words, different carts may not be placed in a strictly sequential way along the moving direction of the conveyor. Therefore, to infer which cart an item belongs to without ambiguity, we need to perform *full-dimension* relative positioning along both X- and Y- axes. *Second*, for ordering purposes, previous work [30] uses Dynamic Time Warping (DTW) algorithm to find the "V-zone" in a tag's phase sequence, which is costly in both time and space. To meet the real-time checkout demand, we wish to reduce the time complexity as much as possible. DTW also requires a reference phase profile generated with a known tag layout as input, which is difficult to obtain in our scenario. *Third*, the RF phase is a sensitive metric that is easily affected by the environment. A more robust approach is in need to further reinforce the ordering accuracy. *Fourth*, even with the above considerations, the ordering results among tags would still contain more or less unanticipated errors, which had better be alleviated.

To address these challenges, we first develop a fast V-zone extraction method for tag ordering along X-dimension, which is time-efficient for its only one run of scanning the whole phase sequence. Then, we propose the concept of *overall phase changing rate* and leverage it to order tags along Y-dimension. Theoretical analyses are given to demonstrate its feasibility (more details will be presented in Section 5). Besides, to combat indoor ambient noise, we figure out an effective solution which combines the measurements of two antennas and utilizes their relative value to mitigate the effect of imperfect signals. Finally, to correctly determine each tag's *host* (*i.e.* the real cart that carries the target tag), we design a holistic item classifying mechanism by incorporating a probabilistic model. The eventual classifying accuracy can then be further enhanced.

1.4 Contributions

This paper presents a comprehensive study of RFID-enabled self-checkout services. Specifically, it makes the following contributions.

- We propose a full-dimension relative positioning schema to essentially improve today's checkout-free experiences (particularly in retail stores). It works based on the phase information acquired from deployed RFIDs, and the accompanying challenges are comprehensively analyzed.
- We design a series of algorithms to accurately determine the tag order in each dimension, as well as to overcome the negative impacts from imperfect measurements, with affordable computation overhead.
- We present a holistic item classifying solution based on probabilistic modeling to map each item to the correct cart that carries it, which can tolerate tag ordering errors to a great extent.
- We implement a prototype system for Taggo with COTS RFID devices and evaluate its performance with extensive experiments. The average ordering and classifying accuracies reach as high as 90% and 95%, which are sufficient for most practical applications.

Roadmap. The rest of the paper is organized as follows. The main design of Taggo is overviewed in Section 3. We introduce the technical details of the tag ordering algorithms along both X- and Y-dimensions in Section 4 and Section 5. The item classifying method of Taggo is elaborated in Section 6. More feasible solutions to enhance Taggo's applicability are presented in Section 7. We describe the implementation and evaluation of our system in Section 8. We review related work in Section 2, and finally conclude this paper in Section 9.

2 RELATED WORK

We briefly review the related literature in this section.

2.1 RF-based Localization

Localization problem has been well studied in the RFID field, which can be mainly categorized into absolute positioning and relative positioning.

Absolute positioning: Early attempts try to use the RSSI information as fingerprint or distance metric for acquiring location information [9, 25, 26, 32, 41, 43]. LandMarc [25] first employs the idea of reference tags with fixed locations to help infer a target tag's location. There are also growing interests in exploiting phase measurements to locate tags. Angle of Arrival (AoA) is a typical solution, which works by measuring the phase difference between the received signals at different antennas [7, 8, 20, 33, 35]. PinPoint [18] proposes a novel algorithm that accurately computes the angle of arrival, allowing multiple collaborating access points to localize interfering transmitters on the order of centimeters even under strong multi-path propagations. The concept of Synthetic Aperture Radar (SAR), which is first used in military radar systems, is also borrowed to the wireless localization domain in recent years [19, 20, 23, 28, 42]. PinIt [34] leverages the SAR technique to extract the multi-path profile of RFID tags for tag localization in non-line-of-sight scenarios. Other approaches try to incorporate different mathematic models to infer the target's location [12, 16, 24, 39]. Tagoram [40] presents the concept of hologram, which successfully handles the thermal noise and device diversity and realizes real-time tracking with high precision in a 2D plane.

Relative positioning: In addition to the absolute location, relative locations among a set of objects are also important in many applications. As pioneer work in RFID-based relative positioning area, OTrack [29] figures out the order of luggage on the conveyor belt using RSSI trends of tags. However, RSSI can not be considered as a reliable metric because it is seriously affected by the indoor multi-path effect, and thus hard to achieve high accuracy. Another standard work in relative positioning is STPP [30], which proposes the concept of V-zone in the phase profile and utilizes it to infer the order of tags. However, it uses the DTW algorithm to extract V-zone from the phase sequence, which is time-consuming, and to apply DTW, it needs to generate a reference phase profile beforehand with known tag layout, which is hard to achieve in many practical scenarios. The authors in

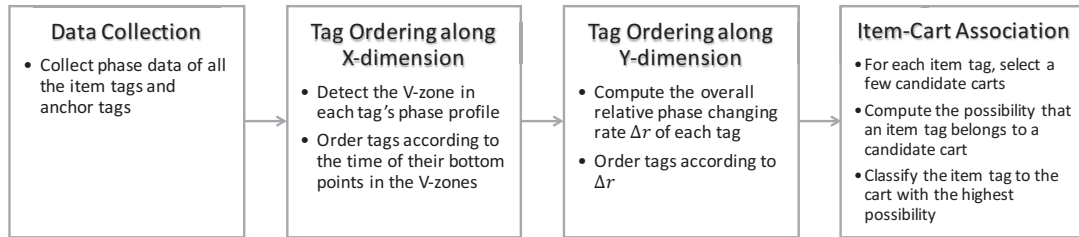


Fig. 2. Workflow overview of Taggo

[22] incorporate a dedicated robot to localize the order of books and detect lying-down books in the library. It still focuses on relative positioning along one dimension.

Our work is inspired by the above works that leverage the RF phase for localization purposes. Although some state-of-the-arts could demonstrate quite small error margins such as [40] and [35], they require dedicated devices (like USRP) or massive deployment costs (for reference tags or antennas) in order to achieve high precision. Other ones demand extra information a priori, including the moving speed of a conveyor ([10]), the relative trajectory of a tag ([28]), or the locations of several antennas ([27]), to generate an (inverse) synthetic aperture that is necessary for localization purposes. However, in our work, centered around the self-checkout context, we aim to provide a more lightweight and efficient schema that is capable of acquiring the order of tags with neither the need to know their absolute locations nor requiring too much prior knowledge.

2.2 Intelligent Self-Checkout Studies

Early researchers in [44] introduce a new form of supermarkets: smart markets, in which customers can acquire detailed product information with easy access and expedite the checkout process without human intervention. A POS (Point of Sale) zone with a shielding door is designed. Each time one user with a shopping cart enters the POS zone, inside which an RFID reader starts to read RFID tags attached to items in the shopping cart.

Recently, the rapid development of computer vision technology facilitates new kinds of self-checkout systems. [38] proposes an intelligent system embedded with a single camera to detect multiple products without any labels (barcodes, RFID tags, or QR codes) in real-time performance. To achieve this, deep learning skill is applied, and data mining techniques construct the image database employed as the training dataset. The authors in [36] analyze the “Just Walk-Out” technology in Amazon Go. With sensors and cameras placed all around the surveillance region, the system can keep track of the products in a virtual cart for each user by utilizing amalgamative technologies of deep learning, computer vision, and sensor fusion.

3 OVERVIEW OF TAGGO

This section briefly describes the workflow of Taggo. As demonstrated in Fig. 1, there is one conveyor belt that could carry multiple users' shopping carts. Different carts may be placed in a back-to-back (*i.e.*, along X-dimension) or side-by-side (*i.e.*, along Y-dimension) manner, and people can also drop the goods they decide not to buy at the last minute onto the belt (of course, outside the carts). There are two types of tags, *i.e.*, *anchor tags* which are attached on four surrounding edges of each cart to represent the cart, and *item tags* which are attached to the goods for identification purpose. Taggo decomposes the self-checkout problem into relative positioning among tags inside the carts, tags outside the carts, and anchor tags on the carts, in order to finally separate those goods that need to be checked out with regard to a specific cart/customer. Without loss of generality, we establish the coordinate system with regard to the plane of the conveyor, and assume the tags/conveyor move(s) along X-axis (*i.e.*, X-dimension) from left to right in our scene. Taggo deploys two reader antennas at each side of the

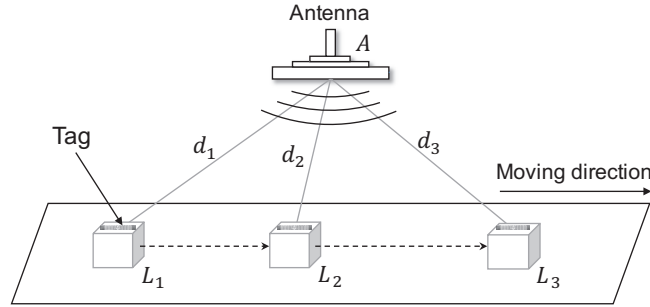


Fig. 3. Relationship of tag and antenna. Tag's distance to the antenna first increases and then decreases.

conveyor to read the tags. Its infrastructure also includes a central server which stores system parameters (e.g. locations of antennas, IDs of the anchor tags), as well as runs the relative positioning algorithms. To accomplish the goal of automatic checkout, Taggo goes through the following steps at a high level:

- *Step 1: Data collection.* We first collect phase data of all the tags (including item tags and anchor tags) with the deployed RFID devices.
- *Step 2: Tag ordering along X-dimension.* With each tag's phase profile, we detect the V-zone with a lightweight method described in Section 4. Then we order tags according to the time when their bottom points in the detected V-zones occur.
- *Step 3: Tag ordering along Y-dimension.* For each tag, we compute an overall relative phase changing rate as proposed in Section 5. Then we order tags according to the computed rate.
- *Step 4: Item-cart association.* For every item tag, we select a few candidate carts with the ordering result along X-dimension in Step 2. Further by combining results in Step 3, we assign a possibility to each candidate cart, which describes how likely it carries the item tag. Finally, the item tag is associated with (or classified to) the cart with the highest possibility.

The whole procedure is summarized in Fig. 2. We will elaborate on the technical details of the above steps in the next few sections.

4 ORDERING ALONG X-DIMENSION

In this section, we first introduce some preliminary knowledge on relative positioning, and then describe how Taggo works to obtain the order of tags along X-dimension.

4.1 V-zone Pattern of Phase Sequence

The RF phase is a basic attribute of a signal and can be reported by commercial RFID readers [11]. Let d be the distance between a pair of tag and antenna. Since the signal traverses a total distance of $2d$ in backscatter communication, the phase rotation output by the reader can be expressed as [15]:

$$\theta = \left(\frac{2\pi}{\lambda} \times 2d + \mu \right) \bmod 2\pi \quad (1)$$

where λ is the wavelength, and the term μ describes the constant phase shift caused by the device's hardware characteristics.

As Fig. 3 illustrates, when a tagged item moves with the conveyor belt along X-dimension, its distance to the reader increases at first until the reader is right above the tag (when AL_2 is perpendicular to L_1L_2), then decreases.

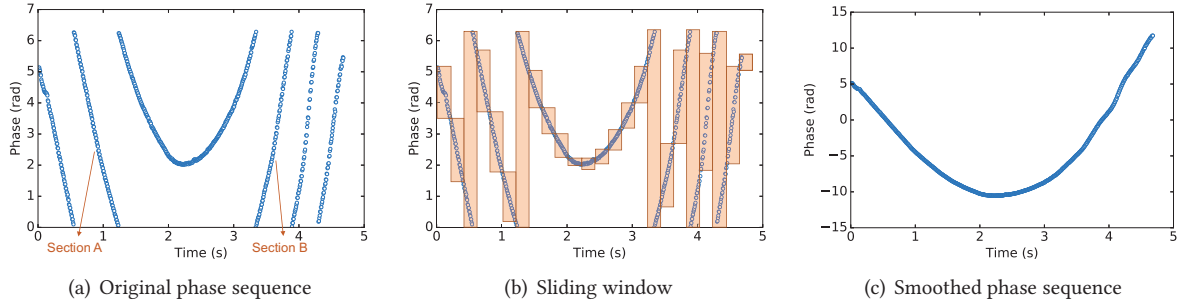


Fig. 4. V-zone pattern of phase profile. (a) An example of the collected phase sequence, which contains a “V-zone” pattern. (b) We search the phase profile using a sliding window. (c) We smooth the original phase sequence by splicing adjacent split parts together.

According to Eqn. 1, the raw phase (without modulo operation) will also first descend to a valley and then go up. In practice, the measured phase value jumps when it approaches 0 or 2π due to the mod operation. Fig. 4(a) shows an example of the collected phase sequence in our experiment. As we can see, there exists a “V-zone” shaped like the alphabet “V” around the valley in the curve. And the bottom of the V-zone, exactly occurs at the time when the reader is closest to the tag. If the moving speed of the tag/conveyor is constant, the phase profile will be symmetric around the bottom point.

Once the V-zone is detected, we are able to tell which tag passes the antenna earlier and which passes later. Tags’ order can then be inferred.

4.2 Limitations of Prior Art

To extract the V-zone in a phase profile, prior work STPP [30] utilizes the Dynamic Time Warping (DTW) technique. A reference phase profile is pre-calculated and the measured phase profile is then matched against the reference one through DTW to find where the V-zone appears. However, DTW has the following two drawbacks. 1) It requires a reference phase profile, which is generated by knowing the layout of tags and reader, and the moving speed of the reader. But in our situation, it is not feasible to assume such prior knowledge when our goal is to figure out the layout of tags. 2) DTW is time-consuming just to search for a V-zone. In STPP, time complexity reaches $O(\frac{NM}{w^2})$, where N and M are lengths of reference and measured phase profile, while w is the length of segment divided into for calculation. w should not be large as a bigger w will cause loss of information in phase profile and reduce the accuracy.

Therefore, to deal with the above limitations and meet the real-time demand in self-service applications, here in this work, we aim to propose a more practical and efficient V-zone detection approach.

4.3 Fast V-zone Detection

How to detect the fixed pattern V-zone in an acquired phase profile with high accuracy and efficiency is a critical task in relative localization. Recall prior work STPP [30], since it moves the reader manually to interrogate the tags, the corresponding phase profile will be stretched because of the unstable moving speed of the reader. That is also the reason why it utilizes DTW to detect a V-zone. Here in our scenario, since we use a uniformly moving conveyor to transport shopping carts and items, the stretch and compression of the phase profile can be negligible. Looking back at Fig. 4(a), we have the following key observation.

OBSERVATION 4.1. *If a V-zone appears in a phase profile, because of modulo operation, the phase values on the two edges of the V-zone approach 2π , which means a jump from 0 to 2π (or 2π to 0) must have happened at such edge point and its adjacent point. And within the whole V-zone, there are no other jumping points except the two edges.*

Motivated by the above observation, to detect a V-zone, we only need to detect the jumping points in phase profile and then check whether the section between two adjacent jumping points conforms to the feature of V-zone.

As we know, the RF phase is a sensitive metric whose value is easily affected by surrounding environment. Thus, as a preprocessing process, we first put a median filter on the raw phase sequence to eliminate noisy points while preserving the lowest point in V-zone. Denote the filtered phase sequence as $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\}$. We use a sliding window w ($w \subset \Theta$) to search from the beginning of Θ . Inside the window, if either of the following conditions is satisfied, we consider w as a *jumping window*:

$$\begin{aligned} \max(w) - \min(w) > \eta \quad \text{or} \\ |last(w) - first(w_{next})| > \eta \end{aligned} \quad (2)$$

where functions $first()$ and $last()$ define the first and last element of w respectively, w_{next} is the window right following w , and η is a pre-defined threshold (we choose η to be π in our experimentation). Fig. 4(b) illustrates an example of sliding windows, we can easily find that inside a window or between windows there may exist a jump, and with the above two conditions such a jump can be detected.

Once we get all the jumping windows and corresponding jumping points inside, according to Observation 4.1, the sections between two consecutive jumping points become the candidates of V-zone. The next step is to find out the real V-zone from all the candidates. We observe when a tag is actually not passing through the antenna, its phase value normally goes up from 0 to 2π or down from 2π to 0 for a V-zone candidate, like section A and B in Fig. 4(a). Then we can filter out these sections with their edge features. Suppose s represents a candidate section. s is determined as a V-zone if the following two conditions are both satisfied:

$$\begin{aligned} 2\pi - first(s) \leq \epsilon \quad \text{and} \\ 2\pi - last(s) \leq \epsilon \end{aligned} \quad (3)$$

Here ϵ is a user-defined small threshold. These conditions ensure that the two edges of the section approach 2π and there is no sharp jump from 0 to 2π or 2π to 0 during this section, which exactly suits the feature of a V-zone. Thus, when Eqn. 3 holds, we can confidently state that the specific candidate section is a V-zone.

Based on the aforementioned principle, we can detect all the V-zones with only one run of scanning the whole phase sequence. To put it simply, when we slide the window w , we first check whether it is a jumping window according to Eqn. 2; then after attaining two consecutive jumping points we further check if the section between them satisfies the edge condition in Eqn. 3 to determine a V-zone. Through this procedure, we achieve a time complexity of $O(N)$ and a space complexity of $O(1)$ for V-zone searching, which is a significant improvement compared against the $O(\frac{NM}{w^2})$ time complexity (almost the square of N) in prior art [30]. This also ensures our approach to work in real-time for self-checkout scenarios.

5 ORDERING ALONG Y-DIMENSION

As we have no restriction on the placement of users' shopping carts, different carts may keep abreast with each other, which means they have similar or overlapped X-coordinates while varies in Y-dimension, as shown in Fig. 1. Therefore, in addition to the X-dimension, we also need to determine the order of tags along Y-dimension.

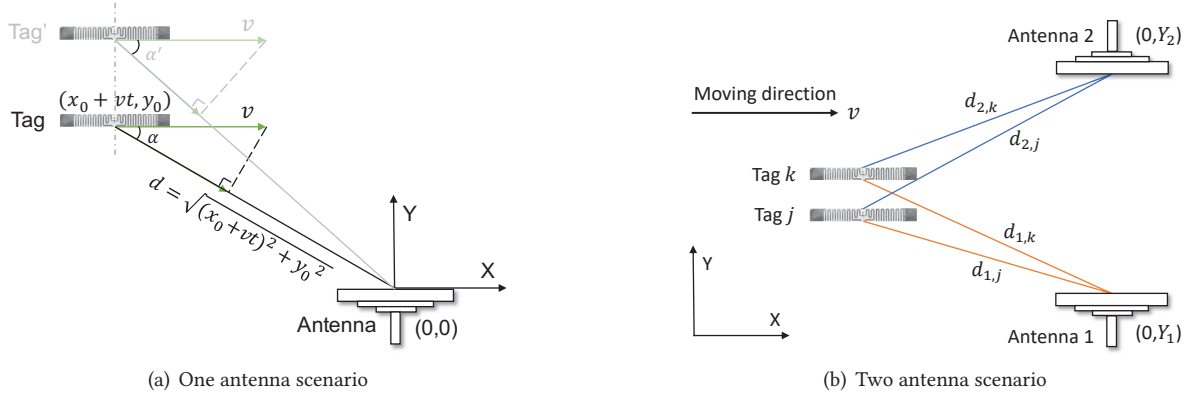


Fig. 5. Geometric relationship between tags and antenna(s). (a) The antenna is located at the origin $(0, 0)$. The tag moves horizontally with speed v from initial position (x_0, y_0) . When a tag is closer to the antenna, it shall observe a larger velocity $v \cos \alpha$ towards the antenna. (b) The coordinates of two antennas are $(0, Y_1)$ and $(0, Y_2)$ respectively. If tag j is closer to antenna 1 than tag k is, then we can easily infer tag j is farther from antenna 2 than tag k is.

5.1 Utilizing One Antenna

We begin to illustrate our design with one antenna adopted. The more complicated two-antenna approach is discussed in the following part.

As mentioned before, the measured phase sequence is split into many short discontinuous parts due to the mod operation. For better illustration, we first smooth the curve by splicing adjacent split sub-sequences together, through the following equation:

$$\theta_i = \begin{cases} \theta_i - \lceil \frac{|\theta_i - \theta_{i-1}|}{2\pi} \rceil 2\pi, & \text{if } \theta_i - \theta_{i-1} > \eta \\ \theta_i + \lceil \frac{|\theta_i - \theta_{i-1}|}{2\pi} \rceil 2\pi, & \text{if } \theta_{i-1} - \theta_i > \eta \\ \theta_i, & \text{otherwise} \end{cases} \quad (4)$$

where $i > 1$. Here the basic rationale is that if the difference between two adjacent phase values exceeds a big threshold η (e.g., $\eta = \pi$), we should compensate an integral multiple of 2π for the current value. An example of the smoothed curve is shown in Fig. 4(c).

We further compare phase profiles of tags with different Y-coordinates, and comes to the following observation:

OBSERVATION 5.1. *When a tag is closer to the antenna in Y-dimension (or in other words, has a smaller Y-coordinate), its phase profile will show a faster changing rate on the whole.*

This observation can be mathematically proven as below. For ease of description, suppose the antenna is located at $(0, 0)$ in the coordinate system, as shown in Fig. 5(a). Let (x_0, y_0) be the initial position of the tag, and v be its moving speed. As the tag moves along X-axis, its X-coordinate $x(t)$ equals $x_0 + vt$ at time t while Y-coordinate $y(t)$ do not change over time, i.e., $y(t) = y_0$. So the distance between the tag and antenna is $d = \sqrt{x^2 + y^2} = \sqrt{(x_0 + vt)^2 + y_0^2}$. Then, we can rewrite the phase expression in Eqn. 1 as below:

$$\theta = \left(\frac{4\pi}{\lambda} \sqrt{(x_0 + vt)^2 + y_0^2} + \mu \right) \bmod 2\pi \quad (5)$$

So the changing rate r of phase over time is theoretically given by

$$r = \frac{d\theta}{dt} = \frac{4\pi}{\lambda} \cdot \frac{v^2 t + vx_0}{\sqrt{(x_0 + vt)^2 + y_0^2}} \quad (6)$$

Apparently, the changing rate r is inversely proportional to tag's Y-coordinate y_0 given a specific time t and initial position x_0 . A larger r means a smaller Y-coordinate, which is also in accord with our observation. We can also understand this from another intuitive perspective. Like Fig. 5(a) shows, as a tag moves horizontally, it will possess a velocity component of $v \cos \alpha$ towards the antenna. So when it situates at a longer distance from the antenna along Y-dimension, a larger α would be observed, and thus a lower $v \cos \alpha$, which further leads to a smaller changing rate in phase.

Based on Eqn. 6, assuming all tags maintain the same x_0 , to obtain the order of tags along Y-dimension, one straight-forward method is to calculate their phase changing rates at a specific time point t_i , and order the results in an inverted manner. To be specific, given a measured phase profile $\{ \langle t_1, \theta_1 \rangle, \langle t_2, \theta_2 \rangle, \dots, \langle t_N, \theta_N \rangle \}$, then the phase changing rate $r(t_i)$ at time t_i ($i > 1$) is computed as

$$r(t_i) = \left| \frac{\theta_i - \theta_{i-1}}{t_i - t_{i-1}} \right| \quad (7)$$

However, this naive method can not directly work in our case because of the following two challenges. First, we know the phase is a sensitive metric and practical measurements are easily affected by surrounding noise even though the tag's position keeps invariant. Hence, it is far from accurate to utilize only one sampled changing rate for tag ordering. Second, also the most important, the assumption that all tags maintain the same x_0 , which means they have the same X-coordinate, can hardly hold in practical situations. We also cannot have such strong constraints on tag's relations along X-axis when our final goal is to order them on the full X-Y dimensions.

Concerning the first challenge, we find that even though the phase value may exhibit some noise at some time points, the overall changing trend of the phase profile agrees with our theoretical analysis. Based on this, we propose to integrate the phase changing rate over a chosen time window, and then exploit the integrated value, which we also call the *overall changing rate*, as a metric for ordering tags. As for the second challenge, without the assumption on x_0 , tags may hold various X-values. Recall Eqn. 6, it is apparent that the phase changing rate is not only determined by the tag's Y-coordinate, but also related to its initial X-position x_0 and time t . Thus, we cannot simply apply a fixed/absolute time window for all the tags as the computed changing rates in this way may not conform to tags' real order. The key issue here is how to choose appropriate time windows that suit different tags.

We already know that all tags' phase profiles demonstrate similar V-zone patterns. Suppose there are total M tags on the conveyor. Then for the j^{th} ($1 \leq j \leq M$) identified tag, we can extract the bottom point of its V-zone leveraging the technique proposed in Section 4. Let τ_j be the time of the bottom point, which exactly depicts when the tag passes the antenna. Formally, we set the time window as the interval below:

$$\begin{aligned} & [\tau_j - T/2, \tau_j + T/2], \\ & 0 < T \leq \min\{2\tau_1, 2\tau_2, \dots, 2\tau_M\} \end{aligned} \quad (8)$$

where T denotes the window size. Intuitively, for each tag, the above equation selects a time interval that is symmetric around the bottom point of its V-zone, as the integrating range. Fig. 6 gives an illustrative example considering two tags. The time window corresponds to an overlapped moving range of the two tags. Eqn. 8 guarantees that the chosen T would suit all the tags. Apparently, the upper bound of T is related to the smallest time τ among all the tags. In our experiment, the carts/tags are driven by a robot with a slow speed (about $0.15m/s$) moving along a line with a length larger than $1m$, so the window size could be a few seconds, which

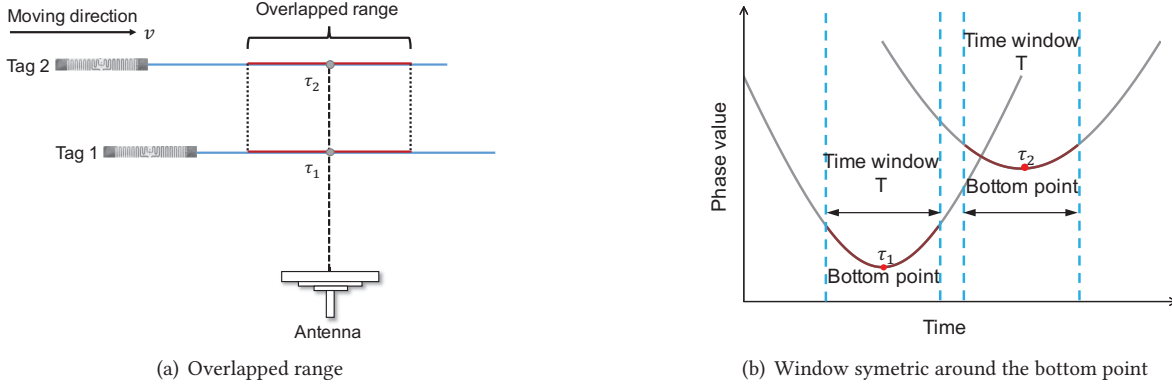


Fig. 6. Choosing a proper time window. (a) When two tags move along the conveyor, they share some overlapped moving range. (b) We integrate tag's phase changing rate over a time window, which is symmetric around the bottom point of V-zone and corresponds to an overlapped moving range.

is sufficient for integration purposes. We then integrate the phase changing rate over the above time window through the following expression:

$$\hat{r}_j = \int_{\tau_j - \frac{T}{2}}^{\tau_j + \frac{T}{2}} r_j(t) dt \quad (9)$$

$r(t)$ is computed with Eqn. 7 for every possible t falling in the chosen time window. Once we get all M tags' overall changing rates $\hat{r}_1, \hat{r}_2, \dots, \hat{r}_M$, by sorting them in descending order we can obtain the order of tags with their Y-coordinate increasing along Y-dimension. The time complexity of our ordering algorithm is $O(N)$.

5.2 Extending to Two-Antenna Scenario

5.2.1 Motivation. It is known that in real indoor environments, due to the multi-path effect and ambient noise, tag's phase profile can be easily affected and deviate from the theoretical pattern. As we utilize the measured phase as a basis to order tags, such imperfect measurements could directly impair the accuracy of our approach. To deal with this problem and increase the final positioning accuracy, one intuitive choice is to deploy multiple antennas along X-axis to form an antenna array. For each antenna, we can repeat our algorithm to output a candidate result of ordering. Further, by incorporating all candidates through a voting schema, the order of tags can be finally determined. However, this straight-forward method requires three antennas at least for voting, which would increase both economic and space costs. Besides, more antennas along the X-axis mean to deploy a longer conveyor belt, so that commodities travel longer time on it, which is also more time-consuming for users' checkout experience.

5.2.2 Our Schema. We propose a lightweight schema that can mitigate the effect of imperfect measurements to a certain extent by using only two antennas.

As Fig. 5(b) depicts, we arrange the two antennas at both sides of the conveyor along Y-axis while facing each other, which means the antennas are placed perpendicular to the conveyor instead of along it. Let $d_{i,j}$ be the distance between the i^{th} ($i = 1, 2$) antenna and the j^{th} ($1 \leq j \leq M$) tag. $r_{i,j}$ denotes the phase changing rate of tag j with respect to antenna i . Given two tags, tag j and tag k , assuming their X-coordinates are the same, if tag j is closer to antenna 1 than tag k is, then we can easily infer that tag j is farther from antenna 2 than tag k is. More formally, we have

$$d_{1,j} < d_{1,k} \ \& \ d_{2,j} > d_{2,k}$$

According to Observation 5.1, the following derivation holds:

$$\begin{aligned}
& d_{1,j} < d_{1,k} \ \& \ d_{2,j} > d_{2,k} \\
\Rightarrow & r_{1,j} > r_{1,k} \ \& \ r_{2,j} < r_{2,k} \\
\Rightarrow & r_{1,j} - r_{2,j} > r_{1,k} - r_{2,k}
\end{aligned} \tag{10}$$

Apparently, the above derivation demonstrates that *for a specific tag, the difference of its phase changing rate between two antennas would be enlarged compared against that when only one antenna is adopted*. So we raise our question: can we use the difference of phase changing rate between two antennas as the metric for tags' order? The answer is yes, and below gives the theoretical proof.

As shown in Fig 5(b), suppose the two antennas are located at $(0, Y_1)$ and $(0, Y_2)$. Without loss of generality, here we assume $Y_2 > Y_1 \geq 0$. Let (x_0, y_0) be the initial position of the tag, and v be its moving speed. Define r_1, r_2 as the phase changing rates reported by the two antennas respectively, and Δr as their difference. Δr is also called *relative phase changing rate*. Similar to Eqn. 6, we have

$$\Delta r = r_1 - r_2 = \frac{4\pi}{\lambda} \left[\frac{v^2 t + vx_0}{\sqrt{(x_0 + vt)^2 + (Y_1 - y_0)^2}} - \frac{v^2 t + vx_0}{\sqrt{(x_0 + vt)^2 + (Y_2 - y_0)^2}} \right] \tag{11}$$

Obviously, the value of y_0 falls in the interval of $[Y_1, Y_2]$ with the assumption $Y_2 > Y_1 \geq 0$. According to the above equation, the relative phase changing rate Δr would decrease monotonically with the increase of tag's Y-coordinate y_0 .

Here in two-antenna scenario, for one specific tag, we have two phase profiles measured by two antennas respectively. Similar to the approach adopted in Section 5.1, for the j^{th} tag, we integrate the relative phase changing rate over a pre-defined time window as below

$$\Delta \hat{r}_j = \int_{\tau_j - \frac{T}{2}}^{\tau_j + \frac{T}{2}} \Delta r_j(t) dt = \int_{\tau_j - \frac{T}{2}}^{\tau_j + \frac{T}{2}} r_{1,j}(t) dt - \int_{\tau_j - \frac{T}{2}}^{\tau_j + \frac{T}{2}} r_{2,j}(t) dt \tag{12}$$

Since we arrange the two antennas along the Y-axis, the bottoms of their V-zones would occur at nearly the same point in time for one tag. So here we can use the same time interval $[\tau_j - T/2, \tau_j + T/2]$ (see Eqn 8) for the two antennas for simplicity. $r_1(t)$ and $r_2(t)$ are also computed through Eqn 7 in a similar way. After we get all M tags' overall relative changing rates $\Delta \hat{r}_1, \Delta \hat{r}_2, \dots, \Delta \hat{r}_M$, by sorting them in descending order we can obtain the order of tags with their Y-coordinate increasing along Y-dimension. In addition to resisting measurement errors, the adoption of two antennas can also reduce the negative effect of tag's height to a certain extent, as we soon will discuss in the later section.

6 SYSTEM WORKFLOW

Remember that our final goal is to determine which cart a tagged item belongs to. So far we have realized tag ordering along both X-dimension and Y-dimension. In reality, sometimes we need not know the exact order of all tags. For example, the order among tags inside the same shopping cart is not so important compared with the order of tags near the anchors, since we mainly focus on the relative position between anchor tags and item tags. Next in this section, we will introduce the overall workflow of Taggo in self-checkout context.

6.1 Acquiring Relationship of Anchor Tags

As Fig. 1 shows, every shopping cart has four anchor tags on its four sides. When placed on the conveyor belt, we have no idea about which two tags are along X-axis and which are along Y-axis. The first thing is to figure out the geometric relation between anchor tags. Without loss of generality, suppose anchor tags are arranged in a clockwise manner, with tag 1 sitting opposite to tag 3 and tag 2 opposite to tag 4. We find that if tags are along

Y-axis, the time they pass through the antenna would be similar, whereas a large difference would appear if tags are along X-axis. Therefore, to infer one group of tags belong to which axis, we utilize the bottom point in phase profile to calculate the following difference:

$$\begin{cases} \delta_1 = \tau_1 - \tau_3 \\ \delta_2 = \tau_2 - \tau_4 \end{cases} \quad (13)$$

As illustrated in Section 5.1, τ_i denotes the time of anchor tag i 's bottom point, which we also call "D-time" for short. If $\delta_1 > \delta_2$, tag 1, tag 3 are supposed to be along X-dimension and tag 2, tag 4 are along Y-dimension, and vice versa. Further, through our proposed ordering method we can also find out the relative positions of tags along the same dimension.

6.2 Classifying Item Tags

Now that we get the geometric relation among anchor tags, to determine which cart an item tag belongs to, the first thing is to obtain the full-dimension relative positions among the tag and all carts' anchor tags. Ideally, the location of the target tag should exactly fall into the region formed by four anchor tags of a specific cart. However, due to the instability of phase measurements and indoor environment, the ordering result we get could involve unknown errors, so that the target tag may not even fall into any feasible cart area. To address this issue, we propose to incorporate a probabilistic model. Our basic idea is if the ordering algorithm infers that one item tag has a similar position with not only one anchor tags, we should include all these anchor tags/carts as candidates.

To be more concrete, let τ be the D-time of the target tag, and $\Delta\hat{r}$ be its overall relative phase changing rate. For every possible cart C_i , suppose it carries four anchor tags I_1, I_2, I_3, I_4 with I_1, I_3 along X-dimension and I_2, I_4 along Y-dimension. Let us first consider the X-dimension. If $\tau_{I_1} < \tau < \tau_{I_3}$, then cart C_i is obviously a candidate host of the item tag. Moreover, if $|\tau - \tau_{I_1}| < \varepsilon$ or $|\tau - \tau_{I_3}| < \varepsilon$ (ε is a pre-defined small threshold), or in other words, the item tag is very close to one of the anchor tags along X-dimension, we should also include C_i as a candidate. By traversing all the carts, suppose we have selected L candidates C_1, C_2, \dots, C_L in total. Then we turn our attention to Y-dimension. For every candidate cart C_i , we give it a score p_i to measure the probability that it indeed carries the target tag. p_i is computed through the following probabilistic model:

$$p_i = \frac{e^{z_i}}{\sum_{l=1}^L e^{z_l}} \quad (14)$$

where

$$z_i = -(|\Delta\hat{r} - \Delta\hat{r}_{I_2}| + |\Delta\hat{r} - \Delta\hat{r}_{I_4}|)$$

p_i is essentially a softmax function, also known as the normalized exponential function. Softmax is a generalization of the logistic function to multiple dimensions, and is commonly used in neural networks to normalize a given vector. $\Delta\hat{r}$ is computed with Eqn. 12. Here, since $z_i = -(|\Delta\hat{r} - \Delta\hat{r}_{I_2}| + |\Delta\hat{r} - \Delta\hat{r}_{I_4}|)$, it is easy to figure out that if a tag indeed belongs to the candidate cart C_i , $\Delta\hat{r}$ would take a value between $\Delta\hat{r}_{I_2}$ and $\Delta\hat{r}_{I_4}$, thus resulting in a relatively big z_i and p_i . A higher p_i also indicates the item tag is more likely to belong to that candidate cart. Eventually, by traversing all the candidate carts, the target tag is supposed to be carried by the cart which has the highest score p_i , or in other words, the tagged item is classified to the cart that has the highest p_i . In case that the target tag is dropped by customers, which means it does not belong to any cart, we judge each computed score with an empirical threshold. If all scores are below this threshold, we infer that the item is a dropped one.

6.3 Considering Item's Height

Our analyses in Section 4 and 5 only consider the X-Y plane, which have an implied assumption that all tags have the same height. But in real-world retail applications, commodities could have quite different heights, and we

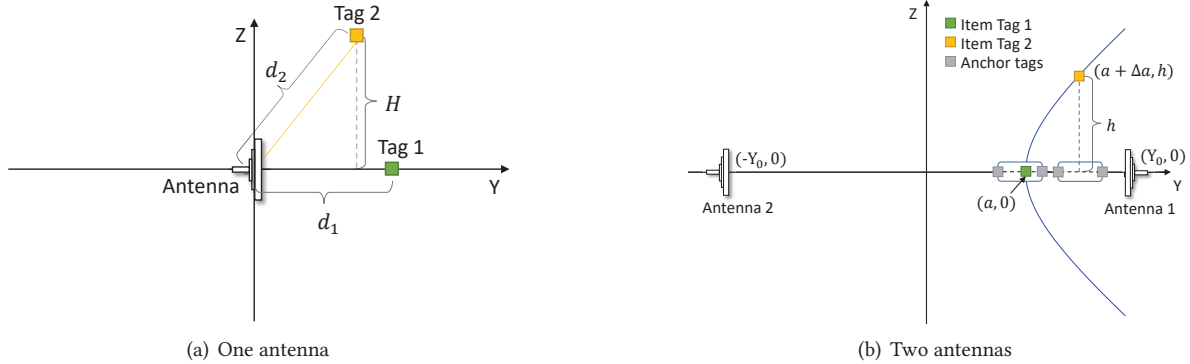


Fig. 7. Ambiguity caused by tag's height. (a) Tag 1 has a bigger Y-coordinate than Tag 2, but Tag 2 is farther to the antenna because it has a larger height "H". (b) For every point on the hyperbola, the difference of its distances to Antenna 1 and Antenna 2 is constant.

can not strictly require users to keep the tags on their purchases along the same height. In this part, we further discuss the influence of tag's height on our result.

Even if different tags may get various heights, their V-zone patterns still hold and the relative time order of their bottom points remains unchanged as long as they maintain the same X-coordinates. So the ordering result along X-dimension is independent of tag's height.

We now consider the Y-dimension. Recall that we use tag's phase changing rate to infer its relative location along Y-dimension, as a larger Y-coordinate means a longer distance between the tag and antenna, which results in a smaller phase changing rate. However, such theory could be untenable if we take tag's height (*i.e.*, Z-coordinate) into account. Fig. 7(a) shows an illustrative example, which gives a side view of Y-Z plane. Tag 1 has a bigger Y-coordinate than Tag 2, but Tag 2 has a height of "H". As a result, compared to Tag 1, Tag 2 is farther to the antenna (*i.e.*, $d_2 > d_1$), and thus gets a smaller phase changing rate. Mathematically, if two adjacent items are Δy apart along Y-axis, then they could be mistakenly ordered if their height difference exceeds Δy .

Since we utilize two antennas to deal with measurement errors, we will demonstrate that this can also mitigate ordering mistakes in Y-dimension caused by tag's height. As described in Fig. 7(b) (Y-Z coordinate system), the green and yellow squares denote two item tags and the grey ones represent anchor tags. For the sake of presentation, here we assume all tags have the same X-coordinate. Suppose Tag 1, Tag 2 and the two antennas are located at $(a, 0)$, $(a + \Delta a, h)$ and $(\pm Y_0, 0)$ respectively. We draw a branch of a hyperbola with Antenna 1 and Antenna 2 as foci and $(a, 0)$ as vertex. In other words, for every point on the hyperbola, the difference of its distances to Antenna 1 and Antenna 2 is fixed, which can be expressed as

$$\frac{y^2}{a^2} - \frac{z^2}{b^2} = 1 \quad (15)$$

where $b^2 = Y_0^2 - a^2$. Theoretically, points on the hyperbola should have identical relative phase changing rate Δr as they always maintain the same relative distance to the two antennas. If Tag 2 is located on the hyperbola, then according to Eqn. 15, its height should satisfy

$$h = \frac{\sqrt{Y_0^2 - a^2}}{a} \sqrt{(2a + \Delta a)\Delta a} \quad (16)$$

In such a case, item Tag 2 could be ambiguously ordered as Tag 1, and thus be classified into the wrong cart. Here gives an illustrative example, if $Y_0 = 100\text{cm}$, $a = 60\text{cm}$, and the spacing Δa between the two item tags equals

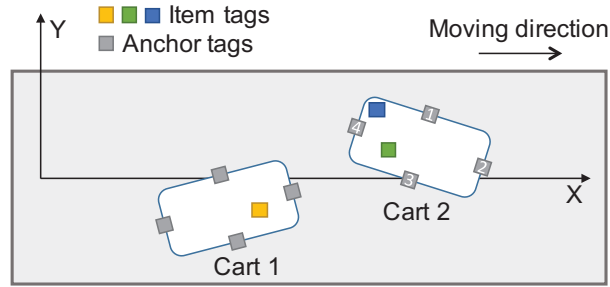


Fig. 8. Scenario where carts are not placed in parallel to the axes. Then item tags inside a cart may not strictly satisfy the relative position constraints with regard to the four anchor tags. The blue item tag may be falsely classified outside carts.

10cm, then h is about 48cm by calculating through the above equation. Namely, Tag 2 should be at least 48cm higher than Tag 1, then their order could be affected. Considering the checkout scenario, this height restriction ($< 48cm$) can be easily met if we guide the users to place their shopping items properly. Besides, since different carts are naturally separated by a distance (often $> 10cm$), this further makes it harder to classify items into false carts. We will give more experimental evaluations in the next section.

7 ENHANCING TAGGO'S PRACTICAL APPLICABILITY

In this section, we present more feasible solutions to enhance Taggo's applicability to various real-world circumstances, and give useful strategies to guide the configurations of crucial thresholds involved in our algorithms.

7.1 Cart Placement

In practical applications, users may not place their carts in the desired manner that is parallel to the moving direction of the conveyor (*i.e.*, X-axis). Then item tags inside a cart may not strictly satisfy the relative position constraints with regard to the four anchor tags. Fig. 8 shows an illustrative example. The blue tag in the figure observes a slightly larger relative position along Y-dimension than the anchor tag 1, while its relative position should lie between anchor tag 1 and 3, in an ideal situation where the cart is placed parallel to X-axis. So the blue item tag could be falsely classified in theory. But recalling Section 6.2, since we have employed a probabilistic model to classify tags by incorporating measurements from both X- and Y-dimension, generally speaking, the blue tag in Fig. 8 still gets more chance to be classified into the correct cart (cart 2). Besides, to better deal with such cases, we suggest that a feasible solution is to deploy an extra pair of anchor tags onto the two diagonal positions of each shopping cart. Then the relative position of an arbitrary item tag inside the cart should definitely lie between the two diagonal anchor tags. We can utilize this constraint to make a further validation on which cart (if any) one item tag belongs to.

7.2 Undesired Settings

Multi-path effect. In indoor environments, there exists multi-path effect, especially when people moving around, making the final received phase a combination of multiple copies of RF signals and thus deviate from the theoretical value. As demonstrated in Section 5.2, we have combined measurements from two properly deployed antennas, which can mitigate the impact from the multi-path effect to a certain extent. In addition, we would suggest utilizing more antennas to further reduce the negative impact from multi-path and enhance the final accuracy.

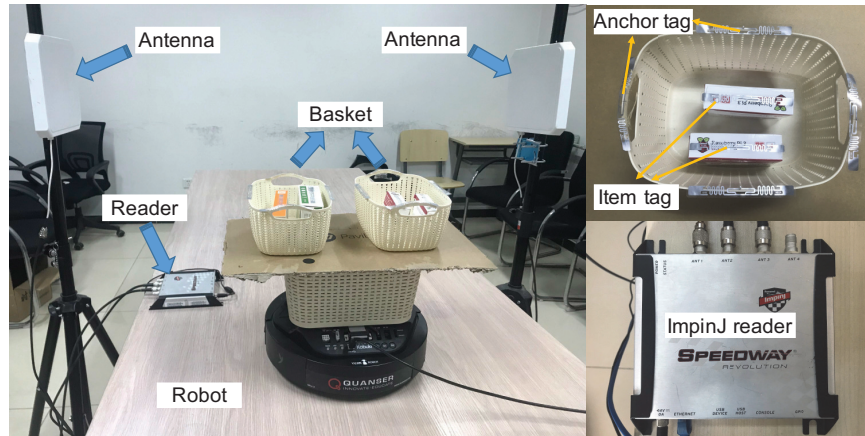


Fig. 9. Experiment setup. We build a prototype and evaluate Taggo using commercial RFID devices.

Liquid/metal objects. If tags are attached on metallic (or liquid) surfaces or surrounded by such objects, their replies will be drawn in the reflections from these materials. Apparently, our solution fails to work if the reader cannot receive replies from tags. In such situations, we would suggest using the anti-metal (or anti-water) RFID tags [14, 21]. These tags are made of special materials like ceramics and trickily designed in circuits. Even when there exist strong interferences from surrounding objects, the reader can still receive tags' replies.

7.3 User-Defined Thresholds

η in Section 4.3 and Section 5.1. As we know, the measured phase has a range of $[0, 2\pi]$, and thus jumps when it approaches 0 or 2π due to the mod operation. The threshold η is used to determine whether there exists such a phase jump. So we should set η to a relatively large value. According to our empirical study, it is feasible that η falls into $[\pi/3, 5\pi/3]$. So without loss of generality, we choose η to be π by default. Besides, as η inherently results from the tag's phase attribute, it is independent of other system settings such as item size and orientation.

ϵ in Section 4.3. The threshold ϵ is used to check whether a phase measurement approaches 2π . So it should be set to a small value. Besides, the measured RF phase also follows a typical Gaussian distribution with a standard deviation of about 0.1 radians, affected by basic thermal noise at the receiver side [40]. Considering this, η should be several times greater than 0.1. With our empirical study, we select ϵ as $\pi/3$ by default.

ϵ in Section 6.2. The threshold ϵ is utilized to determine whether two points are close to each other in time domain. Essentially, our algorithm wants to tolerate the scene that two tags get similar locations along X-dimension. Considering the size of practical shopping carts, we think that if two tags get a distance fewer than 3 centimeters, then they can be regarded to have 'similar' locations. In our experimentation, the tags move at a uniform speed of about 0.15 m/s. Then 3 cm distance corresponds to a time duration of about 0.2 seconds. Thus, a reasonable value of threshold ϵ can be set to 0.2 s. According to the above analysis, ϵ is apparently related to the moving speed of the conveyor that drives tags. If the tags move faster, ϵ should be tuned to a smaller value, and vice versa.

8 IMPLEMENTATION & EVALUATION

We have implemented Taggo using COTS UHF RFID devices and conducted performance evaluation in our lab environment as shown in Fig. 9.

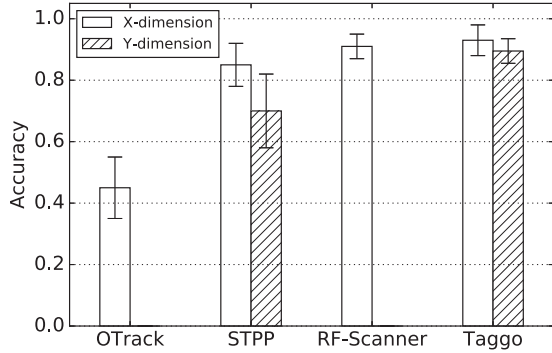


Fig. 10. Ordering accuracy

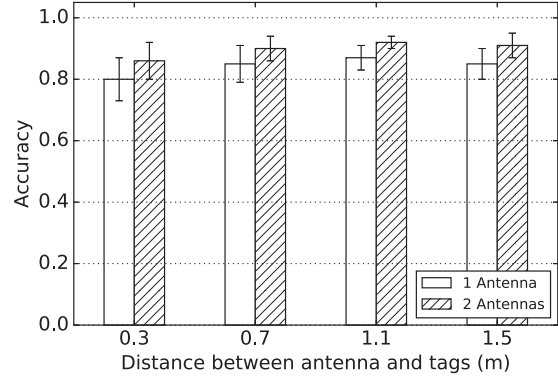


Fig. 11. Accuracy vs. antenna number

8.1 Prototype

Hardware: Taggo adopts an Impinj Speedway Revolution R420 reader [2], compatible with EPC Gen2 standard and operating in the frequency band of 920.5 ~ 924.5MHz by default. The reader is connected to our host end through Ethernet. We employ two antennas with circular polarization and 8dBi gain, whose sizes are 22.5 cm × 22.5 cm × 4 cm. Four types of tags from Alien Corp, modeled “2 × 2”, “Square”, “Squig” and “Squiggle” are employed. Each tag only costs about 7 cents on average [1], which is cost-competitive in the manufacture of commodities.

Software: We adopt the Low Level Reader Protocol (LLRP) [13] to communicate with the reader. Impinj reader extends this protocol for supporting the phase report. We adjust the configuration of the reader to immediately report its readings whenever tags are detected. The software of Taggo is implemented using Java language. We use a MacBook Pro laptop to run all our programs, as well as connecting to the reader under LLRP. The machine equips Intel Core i7 CPU at 16GB memory.

System Settings: In our experimentation, we utilize a robot car to simulate a conveyor belt (see Fig. 9). The mobile robot can be programmable to run in a linear track at different speeds. On the robot car, we install a few plastic baskets, with RFID tags attached on their four sides. Two types of baskets are adopted, with sizes of 29 cm × 21 cm × 12 cm and 22 cm × 15 cm × 10.5 cm respectively. We also put some tagged items inside and outside the baskets on the robot platform. Two antennas are deployed on the two sides of our experimental desktop, perpendicular to the moving track of the robot. Ground truth locations of antennas and tags are measured by a laser range finder with a supposed error of ±0.1mm. Other system settings, such as the distance between baskets, positions of antennas, types of tags, are tunable parameters, which we will evaluate later.

8.2 Ordering Accuracy

The ordering accuracy among tags plays a key role in Taggo’s performance. Note that a tag is correctly localized if and only if the detected order of the tag is equal to the actual order of it. We define the *ordering accuracy* R_o as the ratio below

$$R_o = \frac{\# \text{ of tags ordered correctly}}{\# \text{ of tags in total}} \times 100\%$$

We deploy up to 20 tags with various X- and Y- coordinates on the robot platform and manipulate the robot to move with a uniform speed of about 0.15m/s. We compare Taggo with other four state-of-the-art relative positioning schemas: a) OTrack [29]: leverages RSSI trends and reading rates of tags to infer their order along X-axis; b) STPP [30]: utilizes the spatial-temporal dynamics in phase profiles to infer the order of tags; c) RF-Scanner

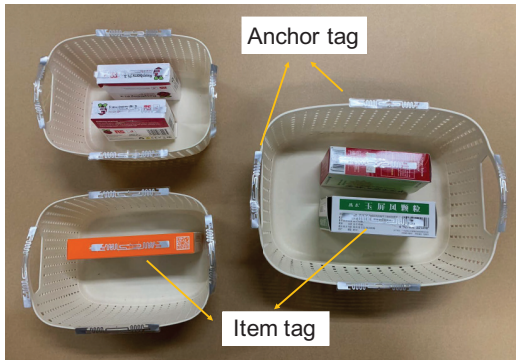


Fig. 12. Baskets are placed in a front-to-back or side-by-side way

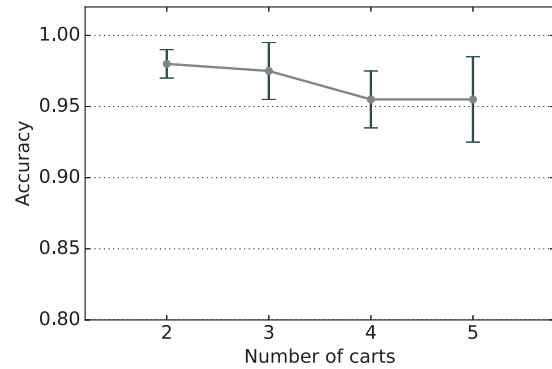


Fig. 13. Classifying accuracy vs. # of baskets

[22]: exploits phase characteristics of RFID tags and incorporate a robot to localize books and detect lying-down books. We conduct experiments under various settings with different numbers of tags and layouts. For each setting, we repeat the experiment 50 times. Fig. 10 plots the average accuracy ratio along both X-dimension and Y-dimension.

X-dimension accuracy: From Fig. 10 we find that Taggo has competitive accuracy with STPP and RF-Scanner, while the accuracy of OTrack is significantly lower than the other three. This is easy to understand because OTrack utilizes RSSI as the ordering metric, which is highly sensitive to multi-path propagation and antenna gain, and thus prone to be unreliable compared against the phase attribute. With the complex DTW method, the precision of STPP can reach 88%. RF-Scanner also achieves an average accuracy of 91% with a self-designed robot component. Compared to OTrack which requires a reference phase profile and RF-Scanner which incorporates dedicated hardware, Taggo is more time-efficient and light-weighted. Overall, our schema achieves a mean ordering accuracy of 93% with a standard deviation of 5% along X-dimension, which is fairly good for most applications.

Y-dimension accuracy: Since OTrack and RF-Scanner only focus on positioning in X-axis, we compare Taggo with STPP which reports Y-dimension results. As shown in Fig 10, Taggo achieves a mean accuracy of 89% with a standard deviation of 4% along Y-dimension. Since we propose to leverage the overall phase changing rate and combine measurements of two antennas to handle negative impacts from environmental noise, Taggo proves to be robust and has very small variances in different settings. The final ordering accuracy in the combined dimension reaches 90% on average.

Effectiveness of utilizing two antennas: To intuitively validate Taggo's performance under two-antenna scenario, we further make a controlled study with only one antenna adopted and compare their results in Fig. 11. We vary the distance between antenna and conveyor from 0.3 m to 1.5 m. It can be seen that in either of these settings, the errors among various distances have little difference. The ordering accuracy along Y-dimension with two antennas increases up to 92% while that with one antenna is only 87%. Clearly, our idea to utilize two antennas is feasible and can directly promote the final accuracy in indoor environments. The straight-line distance between either of the antennas and the conveyor is set to 1.2 m as default in our experiments.

8.3 Classifying Accuracy

Targeting at the self-checkout context, whether an item can be classified into the cart that really carries it with high accuracy is of great importance. For one item tag, it is correctly classified if and only if the detected host of

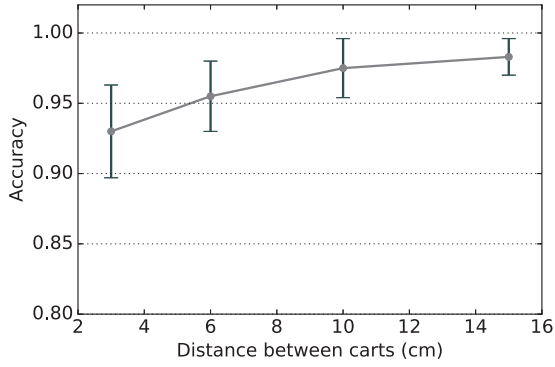


Fig. 14. Classifying accuracy vs. basket distance

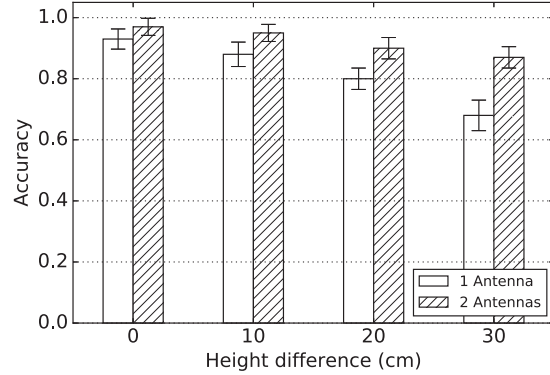


Fig. 15. Classifying accuracy vs. item height

the tag is identical to the real host of it. Consequently, we define the *classifying accuracy* R_c as below

$$R_c = \frac{\# \text{ of item tags classified correctly}}{\# \text{ of item tags in total}} \times 100\%$$

To evaluate Taggo's classifying accuracy, we deploy several plastic baskets on the experimental platform. Some of these baskets are placed in a front-to-back manner along X-dimension and some are placed side-by-side along Y-dimension (see Fig. 12). Anchor tags are attached on the edges of each basket. We run our classifying algorithm mentioned in Section 6 to give predictions on item tags' hosts, and compute the corresponding classifying accuracy.

Number of baskets. To simulate real physics in checkout circumstance, we first vary the number of baskets from 2 to 5, and study the influence from basket population. Fig. 13 presents the average classifying accuracies with different numbers of baskets. We observe that the mean errors among various basket populations are slightly different, from the minimum of 95% to the maximum of 98%. And the result is more errorless when the basket population is small. This is reasonable because interference among baskets would be larger when there are more baskets.

Distance between baskets. Then, to check Taggo's effectiveness when baskets are compactly placed, we change the distances between adjacent baskets from 3 cm to 15 cm with four levels while keeping the same basket population. As shown in Fig. 14, the average classifying accuracies are 93%, 95%, 97% and 98% when basket intervals are 3 cm, 6 cm, 10 cm and 15 cm respectively. We observe from the figure that the mean error degrades a little with the interval between baskets decreases. This is easy to understand because when two adjacent baskets get too close, an item tag between them would be more likely to be mistakenly classified.

Item height. We further study the effect of item's height on Taggo's classifying accuracy. We vary the height of different items and make the maximum height difference between tags as 0, 10 cm, 20 cm and 30 cm respectively. We also utilize one and two antennas to perform experiments to make a comparison study. Fig. 15 plots the final item classifying accuracies. The mean accuracies are 97%, 95%, 90% and 87% respectively in the case with two antennas, while decrease to 93%, 88%, 80% and 68% in the case with only one antenna. We also come to the following two findings from our evaluation results. 1) With the height difference becoming larger, the accuracies in both cases drop to a certain extent. It is reasonable because when there exist higher items, adjacent tags become more easier to be ambiguously ordered. 2) The overall performance in two-antenna case is apparently better than that in single antenna case, which means our approach with two antennas can reduce the negative impact from item height effectively.

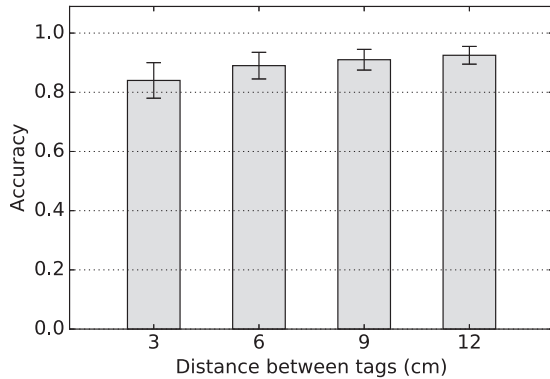


Fig. 16. Impact of tag distance

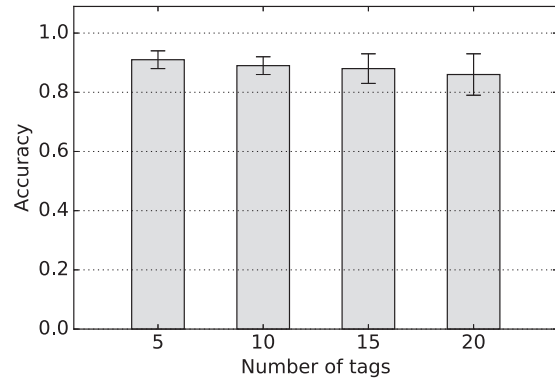


Fig. 17. Impact of tag population

Tag orientation. As commodities in shopping carts may be placed randomly in real-world scenarios, then tags attached may have various orientations. Here we define the orientation of a tag as the angle between its polarization direction and the X-axis. We conduct four groups of experiments with two different types of tags (model “2×2” and “Squiggle”) in a less controlled environment where item tags could maintain different orientations. In the four settings, we make the biggest orientation among item tags as 30°, 60°, 90° and 150° respectively. Table 1 compares the averaged classifying accuracies with regard to tag orientations. We get the following two observations from this table. 1) For the same tag model, the orientation has a slight impact on the final classifying accuracy, and generally speaking, when tag’s orientation difference gets bigger, the accuracy decreases a little bit. This is understandable because it has been demonstrated that tag’s orientation would have a certain effect on its measured phase even though its location remains invariant [17, 37]. But in our checkout scenario, although different item tags may get diverse orientations, when they move with the conveyor, their orientations actually do not change with time. So the impact of tag orientation is very small. 2) For different tag types, “2×2” tends to be more robust to orientation than “Squiggle”. This can be explained by the shape of tags, because 2×2 has a square shape while Squiggle is rectangle. So Squiggle is easier to be affected by the orientation.

To summarize, the overall classifying accuracy of Taggo reaches as high as 95% on average, which is noticeably better than the ordering accuracy. This validates the efficiency of our proposed classifying schema.

8.4 Tuning Parameters

We further discuss the following factors that may have an influence on Taggo’s performance. We mainly focus on the metric of ordering accuracy in this part.

8.4.1 Tag Distance. We first examine the effect from distance between adjacent tags. We range the interval from 3 cm to 12 cm with a step length of 3 cm and carries experiments in each setting. As revealed in Fig. 16, when adjacent tags are placed very close (e.g., 3 cm apart), the mean accuracy drops to only 84%. And with their

Table 1. Classifying accuracy vs. tag orientation

Maximum tag orientation	30°	60°	90°	150°
Accuracy (Squiggle)	97%	96%	95%	96%
Accuracy (2×2)	97%	97%	97%	96%

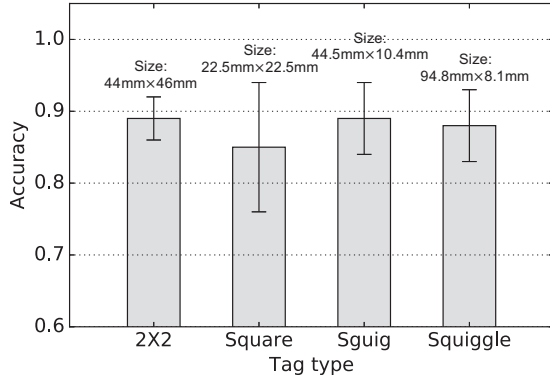


Fig. 18. Impact of tag diversity

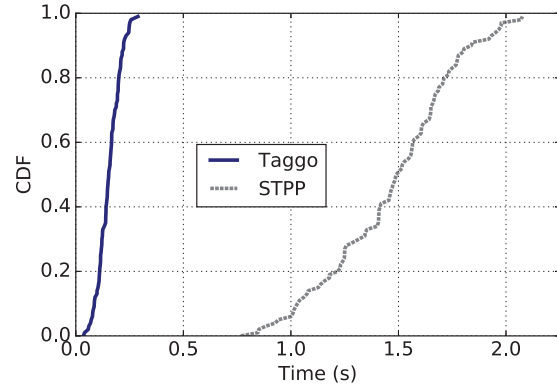


Fig. 19. System latency

distance increasing, the accuracy also shows a rising trend, which is also consistent with our expectation. When tags are placed 12 cm apart, the mean accuracy reaches 92%.

8.4.2 Tag Population. In practical deployments, there could be multiple tagged items in one cart. We then evaluate Taggo’s performance under multi-tag scenario. We change the number of tags from 5 to 20 within the reading zone of antennas. The distance between adjacent tags is randomly chosen from the range of [3 cm, 10 cm]. The results are plotted in Fig. 17. We have the following observations from the figure: a) When there are a small number of tags detected in the reading zone, *e.g.*, 5 tags, Taggo achieves excellent performance with ordering error less than 10%. b) With the tag population gets bigger, the error also grows gradually, to 86% when there are 20 tags. c) Even when there are an adequate number of tags, *e.g.*, over 20 tags, the performance of Taggo still maintains at a moderate level, with about 86% accuracy.

8.4.3 Tag Diversity. We experiment on four models of tags, namely “2 × 2”, “Square”, “Squig” and “Squiggle” to study Taggo’s robustness when different types of tags are adopted. All these tag types have diverse antenna sizes and shapes as depicted in Fig.18. For each tag model, the result is averaged from 50 experiments with the same setting. We find that although the errors of all models maintain at a small value (less than 15%), there exist some differences among them. To be specific, 2×2, Sguig and Squiggle have very close accuracy (*i.e.*, 89%, 89% and 88% respectively), while the Square model observes a lower accuracy of 85% with a higher standard deviation of 9%. This can be explained by the size of tag’s antenna, because Square has a more compact volume (with a size of only 22.5 mm × 22.5 mm) compared with the other three types. Generally speaking, the tag with a larger antenna could absorb more energy from the reader, making its backscattered signal stronger (*i.e.*, higher SNR). The ordering accuracy is thereby higher. In our experimentation, we choose the “Squiggle” model as default.

8.4.4 System Latency. In previous sections, we analytically suggest that the time complexity of Taggo is $O(N)$ along both X- and Y- dimensions. Now we conduct experimental measurements to examine the latency of our system. Total 100 trials are carried out and we compare the ordering latency of Taggo with STPP. The CDF of the result is plotted in Fig. 19. The mean time consumed in Taggo is 0.15 s, about 10× faster than STPP which achieves a latency of 1.47 s on average. Besides, for Taggo, the standard deviation of system latency is 0.05 s, and 90% of the ordering is finished within 0.22 s. Generally speaking, the accuracy and efficiency of our system are high enough to meet most demands in relative positioning domain.

9 CONCLUSION

In this work, we present a lightweight and efficient self-checkout system based on commercial RFID products. Our key innovation is to deploy a few anchor tags on the four edges of each shopping cart and perform relative positioning among tagged items and anchor tags without knowing their absolute locations. Putting our idea into practice, we propose a full-dimension relative positioning schema utilizing the phase profile acquired from RFID tags, and design a holistic item classifying mechanism to assign each item to the correct cart in a probabilistic way. Experimental results demonstrate that Taggo can achieve fairly high accuracy and efficiency while maintaining robustness in various settings. We believe our system will promote more possibilities of the RFID-enabled self-checkout solution in real deployments.

ACKNOWLEDGMENTS

This research is supported in part by the National Key R&D Program of China under grant 2018YFB1004700, the National Natural Science Foundation of China under grants 61902212, 61822205, 61632020 and 61632013, the China Postdoctoral Science Foundation under grant 2019M650683, and the Beijing National Research Center for Information Science and Technology.

REFERENCES

- [1] 2019. Alien Tag Family Product Brief. <http://www.alientechnology.com/wp-content/uploads/Tag>.
- [2] 2019. Impinj Speedway R420 RFID Reader. <https://www.impinj.com/platform/connectivity/speedway-r420/>.
- [3] 2019. Self-Checkout Technology. <http://web.mit.edu/2.744/www/Project/Assignments/humanUse/aychen/references.html>.
- [4] 2020. Fujitsu Demonstrates New Technologies at NRF That Address Pain Point for Retailers: Improve Retail Self-Checkout Experience. <https://www.fujitsu.com/us/about/resources/news/press-releases/2019/ffna-20190110.html>.
- [5] 2020. Nordic ID has set out to eliminate checkout queues - releases ground-breaking solution at NRF. <https://www.nordicid.com/resources/news/nordic-id-has-set-out-to-eliminate-checkout-queues-releases-ground-breaking-solution-at-nrf/>.
- [6] 2020. RFID Based Walk-through Checkout Solution for Future Retail. <https://news.panasonic.com/global/topics/2018/55288.html>.
- [7] Salah Azzouzi, Markus Cremer, Uwe Dettmar, Thomas Knie, and Rainer Kronberger. 2011. Improved AoA Based Localization of UHF RFID Tags Using Spatial Diversity. In *Proceedings of the IEEE International Conference on RFID-Technologies and Applications*. 174–180.
- [8] Salah Azzouzi, Markus Cremer, Uwe Dettmar, Rainer Kronberger, and Thomas Knie. 2011. New Measurement Results for the Localization of UHF RFID Transponders Using an Angle of Arrival (AoA) Approach. In *Proceedings of the IEEE International Conference on RFID*. 91–97.
- [9] Mathieu Bouet and Aldri L Dos Santos. 2008. RFID tags: Positioning principles and localization techniques. In *Proceedings of the 1st IFIP Wireless Days*. 1–5.
- [10] Alice Buffi, Paolo Nepa, and Fabrizio Lombardini. 2015. A Phase-Based Technique for Localization of UHF-RFID Tags Moving on a Conveyor Belt: Performance Analysis and Test-Case Measurements. *IEEE Sensors Journal* 15, 1 (2015), 387–396.
- [11] Daniel M Dobkin. 2012. *The RF in RFID: UHF RFID in Practice*. Newnes.
- [12] Chunhui Duan, Xing Rao, Lei Yang, and Yunhao Liu. 2017. Fusing RFID and Computer Vision for Fine-Grained Object Tracking. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. 1–9.
- [13] EPCglobal. 2010. Low Level Reader Protocol (LLRP).
- [14] Yejun He and Huaxia Zhang. 2013. A New UHF Anti-Metal RFID Tag Antenna Design With Open-Circuited Stub Feed. In *Proceedings of the IEEE International Conference on Communications (ICC)*. 5809–5813.
- [15] Impinj. 2010. Speedway Revolution Reader Application Note: Low Level User Data Support.
- [16] Nanda Gopal Jeevarathnam and Ismail Uysal. 2018. Grid-Based RFID Localization Using Tag Read Count And Received Signal Strength. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 1–8.
- [17] Chengkun Jiang, Yuan He, Xiaolong Zheng, and Yunhao Liu. 2018. Orientation-Aware RFID Tracking with Centimeter-Level Accuracy. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*. 290–301.
- [18] Kiran Raj Joshi, Steven Siying Hong, and Sachin Katti. 2013. PinPoint: Localizing Interfering Radios. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 241–253.
- [19] Rainer Kronberger, Thomas Knie, Roberto Leonardi, Uwe Dettmar, Markus Cremer, and Salah Azzouzi. 2011. UHF RFID Localization System Based on a Phased Array Antenna. In *Proceedings of the IEEE International Symposium on Antennas and Propagation (APSURSI)*. 525–528.

- [20] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. 2014. Accurate Indoor Localization with Zero Start-up Cost. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*. 483–494.
- [21] Laijun Li and Wang He. 2018. Anti-Water UHF RFID Tag Antenna with Multi-Loop Structure for Impedance Matching. In *Proceedings of the Progress in Electromagnetics Research Symposium (PIERS-Toyama)*. 1805–1808.
- [22] Jia Liu, Feng Zhu, Yanyan Wang, Xia Wang, Qingfeng Pan, and Lijun Chen. 2017. RF-Scanner: Shelf Scanning with Robot-assisted RFID Systems. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. 1–9.
- [23] Robert Miesen, Fabian Kirsch, and Martin Vossiek. 2013. UHF RFID Localization Based on Synthetic Apertures. *IEEE Transactions on Automation Science and Engineering* 10, 3 (2013), 807–815.
- [24] Andrea Motroni, Paolo Nepa, Valerio Magnago, Alice Buffi, Bernardo Tellini, Daniele Fontanelli, and David Macii. 2018. SAR-Based Indoor Localization of UHF-RFID Tags via Mobile Robot. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 1–8.
- [25] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil. 2003. LANDMARC: Indoor Location Sensing Using Active RFID. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 407–415.
- [26] Lionel M. Ni, Dian Zhang, and Michael R. Souryal. 2011. RFID-Based Localization and Tracking Technologies. *IEEE Wireless Communications* 18, 2 (2011), 45–51.
- [27] Pavel V. Nikitin, Rene Martinez, Shashi Ramamurthy, Hunter Leland, Gary Spiess, and K. V. S. Rao. 2010. Phase Based Spatial Identification of UHF RFID Tags. In *Proceedings of the IEEE International Conference on RFID*. 102–109.
- [28] Andreas Parr, Robert Miesen, and Martin Vossiek. 2013. Inverse SAR Approach for Localization of Moving RFID Tags. In *Proceedings of the IEEE International Conference on RFID*. 104–109.
- [29] Longfei Shangguan, Zhenjiang Li, Zheng Yang, Mo Li, Yunhao Liu, and Jinsong Han. 2014. OTrack: Towards Order Tracking for Tags in Mobile RFID Systems. *IEEE Transactions on Parallel and Distributed Systems* 25, 8 (2014), 2114–2125.
- [30] Longfei Shangguan, Zheng Yang, Alex X. Liu, Zimu Zhou, and Yunhao Liu. 2015. Relative Localization of RFID Tags using Spatial-Temporal Phase Profiling. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 251–263.
- [31] Christian Tellkamp, Thomas Wiechert, Frédéric Thiesse, and Elgar Fleisch. 2006. The Adoption of RFID-based Self-Check-Out-Systems at the Point-of-Sale. In *Proceedings of the IFIP International Conference on e-Commerce, e-Business, and e-Government (I3E)*. 153–165.
- [32] Chong Wang, Hongyi Wu, and Nian-Feng Tzeng. 2007. RFID-Based 3-D Positioning Schemes. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. 1235–1243.
- [33] Jue Wang, Fadel Adib, Ross Knepper, Dina Katabi, and Daniela Rus. 2013. RF-Compass: Robot Object Manipulation Using RFIDs. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*. 3–14.
- [34] Jue Wang and Dina Katabi. 2013. Dude, Where’s My Card? RFID Positioning That Works with Multipath and Non-Line of Sight. In *Proceedings of the ACM SIGCOMM Conference*. 51–62.
- [35] Jue Wang, Deepak Vasishth, and Dina Katabi. 2014. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. 235–246.
- [36] Kirti Wankhede, Bharati Wukkadada, and Vidhya Nadar. 2018. Just Walk-Out Technology and its Challenges: A Case of Amazon Go. In *Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA)*. 254–257.
- [37] Teng Wei and Xinyu Zhang. 2016. Gyro in the Air: Tracking 3D Orientation of Batteryless Internet-of-Things. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*. 55–68.
- [38] Bing-Fei Wu, Wan-Ju Tseng, Yung-Shin Chen, Shih-Jhe Yao, and Po-Ju Chang. 2016. An Intelligent Self-Checkout System for Smart Retail. In *Proceedings of the International Conference on System Science and Engineering (ICSSE)*. 1–4.
- [39] Fu Xiao, Zhongqin Wang, Ning Ye, Ruchuan Wang, and Xiang-Yang Li. 2018. One More Tag Enables Fine-Grained RFID Localization and Tracking. *IEEE/ACM Transactions on Networking* 26, 1 (2018), 161–174.
- [40] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. 2014. Tagoram: Real-Time Tracking of Mobile RFID Tags to High Precision Using COTS Devices. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*. 237–248.
- [41] Zheng Yang, Chenshu Wu, and Yunhao Liu. 2012. Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*. 269–280.
- [42] Yimin Zhang, Moeness G Amin, and Shashank Kaushik. 2007. Localization and Tracking of Passive RFID Tags Based on Direction Estimation. *International Journal of Antennas and Propagation* 2007 (2007).
- [43] Yiyang Zhao, Yunhao Liu, and Lionel M. Ni. 2007. VIRE: Active RFID-based Localization Using Virtual Reference Elimination. In *Proceedings of the IEEE International Conference on Parallel Processing (ICPP)*. 56–56.
- [44] Fangwei Zheng, Jeffrey Huang, and Mark Meagher. 2009. The Introduction and Design of a New Form of Supermarket: Smart Market. In *Proceedings of the International Symposium on Information Engineering and Electronic Commerce*. 608–611.