Leveraging Read Rates of Passive RFID Tags for Real-Time Indoor Location Tracking

Da Yan, Zhou Zhao and Wilfred Ng
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{yanda, zhaozhou, wilfred}@cse.ust.hk

ABSTRACT

RFID (radio frequency identification) technology has been widely used for object tracking in many real-life applications, such as inventory monitoring and product flow tracking. These applications usually rely on passive RFID technologies rather than active ones, since passive RFID tags are more attractive than active ones in many aspects, such as lower tag cost and simpler maintenance.

RFID technology is also important for indoor location tracking systems that require high degree of accuracy. However, most existing systems estimate object locations by using active RFID tags, which usually incur localization error of more than one meter. Although recent studies begin to investigate the application of passive tags for indoor location tracking, these methods are far from deployable and research of this application is still in its infancy.

In this paper, we propose a new indoor location tracking system, named PassTrack, which relies on the read rates of passive RFID tags for location estimation. PassTrack is designed to tolerate noise arising from external environmental factors, by probabilistically modeling the relationship between tag read rate and tag-reader distance, and updating the model parameters based on the current readings of reference tags.

Besides tolerance of noise, PassTrack is also outstanding in terms of localization accuracy and efficiency. Several new approaches for location inference are supported by PassTrack, and the best one incurs an average error of around 30 cm, and is able to carry out over 7500 location estimations per second on an ordinary machine. Furthermore, as a result of using passive RFID tags, PassTrack also enjoys the many other benefits of passive RFID tags mentioned before. We have conducted extensive experiments on both real and synthetic datasets, which demonstrate that our PassTrack system outperforms the previous localization approaches in localization accuracy, tracking efficiency and space applicability.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems

Keywords

Passive, RFID, tag, read rate, localization

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

1. INTRODUCTION

A variety of methods have been proposed for the purpose of object location tracking, such as those based on GPS, ultrasound, infrared and RFID [10]. While GPS-based localization systems are widely utilized for outdoor environments, the GPS technology has poor performance for indoor applications, due to its requirement of line-of-sight signal reception from the satellites. For indoor location tracking, RFID-based localization technologies have become popular due to the simplicity of attaching tags to target objects, as well as their better tolerance to errors.

There has been over a decade of research on RFID localization, most of which make use of active tags [1, 9, 14] due to their long read range of up to 100 meters. Recently, passive RFID tags are advocated in several studies [3, 6, 10] for location estimation, since they are more attractive than active ones in the following aspects:

- **Lower cost.** The cost of a passive tag is significantly lower than that of an active tag, and therefore passive tags are suitable for those applications where tracking tags are used in a one-off manner. (e.g. inventory monitoring and product flow tracking).

- **Easier maintenance.** Active tags are battery-powered and thus have limited lifetime. For a system using a large set of active tags, the long term maintenance process (e.g. battery replacement) can be complex and costly.

- **Smaller error.** Localization systems that are based on active tags usually incur meters of error [1], which is too large for some indoor applications (e.g. resident monitoring in an elderly house).

Despite the above benefits, most existing research and applications on passive RFID tags simply focus on determining the general existence or non-existence information within the range of RFID detection [2, 3, 5, 6, 8], and research on location estimation using passive tags is still in its infancy.

However, high-accuracy object location estimation is desired for many applications, such as tracking equipment and personnel in hospitals, and providing location-specific information in supermarkets, museums and libraries. For example, the PlaceLab Couple 1 Dataset[1] records the RFID tagging data about a one month stay of a couple in the PlaceLab. In this application, the activities "using a computer" and "sleeping deeply" of a subject cannot be differentiated by an active-tag-based localization system, if the bed is only one meter away from the computer desk. Other applications

[1] `http://architecture.mit.edu/house_n/data/PlaceLab/PLCouple1.htm`

include using the estimated trajectory data for trajectory mining using existing algorithms [15, 16, 17, 18] which can tolerate a certain degree of inaccuracy.

In this paper, we propose a passive RFID system for real-time indoor location tracking, named PassTrack, which translates streams of raw RFID readings into estimated object trajectories with high efficiency. This is important for many database applications involving RFID data.

Our main contributions are summarized as follows:

- Novelty: We are the first to model the relationship between tag read rate and tag-reader distance by a sigmoid curve, while earlier studies use over-simplistic models such as piecewise constant functions [3]. This reader detection model is integrated into a sound probabilistic inference model that enables efficient location inference.
- Noise Tolerance: PassTrack learns the reader detection models online, and is thus able to dynamically adapt the detection model to the changing environment.
- Localization accuracy: We propose several new algorithms for indoor location tracking based on the read rates of passive RFID tags. Our system estimates the location of a constantly moving object (or a static object) with an average error of around 30cm (or below 20cm), which is several times smaller than that of the LANDMARC system [1] that uses active tags.
- Tracking efficiency: Our most accurate algorithm (and thus the slowest due to its heavier computational overhead) is able to perform over 7500 location estimations per second on an ordinary computer, which is still orders of magnitude faster than the sampling-based method of [3].
- Space applicability: We demonstrate that our approach can estimate object locations in a much larger area than that reported in the state-of-the-art work of [10].

The rest of the paper is organized as follows: Section 2 introduces the components of the PassTrack system, including the adaptive reader detection model and the likelihood-based location inference model. The former is further described in Section 3, and the latter is explained in detail in Section 4. In Section 5, we report extensive experiments which verify that our approach is superior to the existing methods in both localization accuracy and efficiency. Finally, we review the related work in Section 6, and conclude the paper in Section 7.

2. PASSTRACK SYSTEM

The setting of our tracking system is outlined as follows: We use n RFID readers/antennae $\{R_1, R_2, \ldots, R_n\}$, m passive reference tags $\{T_1, T_2, \ldots, T_m\}$, and u passive tracking tags $\{O_1, O_2, \ldots, O_u\}$ to track u objects. The key to our approach is to decouple the *reader detection model* from the location inference process:

- For each reader/antenna R_i, its detection model is "learned" from the current read rate of the reference tags dynamically, where the read rates are determined from the readings.
- Using the learned reader detection models, for each tracking tag O_i, we find its most likely location based on the observed read rate of O_i from each reader.

This decoupling strategy is important for PassTrack. While external factors such as changes in temperature and humidity, and the number of objects nearby may influence the detection performance of a reader, our approach is able to dynamically adapt the detection model to the changing environment, due to the online learning of the reader detection models. We elaborate on these parts in the following two sections.

3. READER DETECTION MODEL

3.1 Reader Detection Model Learning

The location inference model of PassTrack relies on the *reader detection model*, which estimates the distance of a tag to an RFID antenna, by the readings of that tag received by the antenna. RSSI (received signal strength indicator), which is widely adopted in the active-tag-based systems [1, 9, 11, 12, 14], is usually not available for passive tags [10]. Therefore, we leverage the read rate information for distance estimation.

The read rate of a tag by an antenna is usually estimated by the tag response count in a fixed number of interrogation cycles sent from the antenna. For example, if an antenna detects the response from a tag in 3 out of a series of 10 interrogation cycles, the read rate is estimated to be 0.3.

We formally define the reader detection model as follows:

DEFINITION 1 (READER DETECTION MODEL). *The reader detection model of a reader/antenna R is a function $p(\ell)$ such that, given the Euclidean distance $\ell = ||T, R||$ between a tag T and the reader R, returns the expected read rate of T by R.*

Both [2] and [3] observed that the read rate of a passive tag remains high (close to 1) when the tag is within a certain distance δ_{major} from the reader, and then decreases almost linearly with the increment of the distance until the read rate reaches 0, where the corresponding distance is δ_{minor}. The former range (0 to δ_{major}) is termed the *major detection region*, while the latter (δ_{major} to δ_{minor}) is termed the *minor detection region*.

The shape of the "read rate vs. distance" curve is like an upside-down curve of the logistic function $f(\ell) = \frac{1}{1+e^{-\ell}}$, where ℓ is the distance between the tag and the antenna. Therefore, we choose the function $p(\ell) = \frac{1}{1+e^{a\ell+b}}$ $(a > 0, b < 0)$ to fit the reader detection model, where a can be adapted to control how sharp the read rate decreases in the *minor detection region*, and b can be adapted to control the range of the *major detection region*.

Now, we formulate our reader detection model for reader R_i as follows:

$$p_i = \frac{1}{1 + e^{a_i \ell_i + b_i}} \ (a_i > 0, b_i < 0). \tag{1}$$

where p_i is the tracking/reference tag read rate at R_i, and ℓ_i represents the distance from the tracking/reference tag at location $\vec{l} = (x, y)$, to reader R_i at location (x_i, y_i), and is given by:

$$\ell_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}. \tag{2}$$

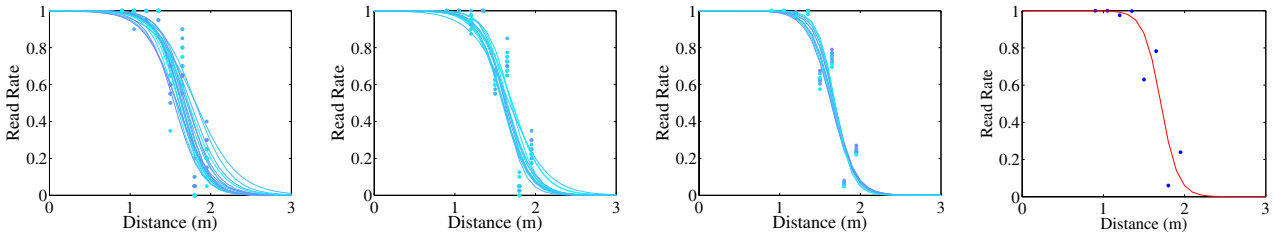
For each reader R_i, there are two parameters (a_i and b_i) that can be determined dynamically. This allows for the adaption of the model to a changing environment. We next explain how to estimate the current model parameters a_i and b_i, based on the read rates of the reference tags.

Let p_{ij} denote the current read rate of each reference tag T_j estimated by reader R_i. Furthermore, since the locations of the readers and the reference tags are fixed, we can easily get the distance $\ell_{ij} = ||R_i, T_j||$ between R_i and T_j. As a result, during each time step, we

(a) Experimental Setting on the Ground

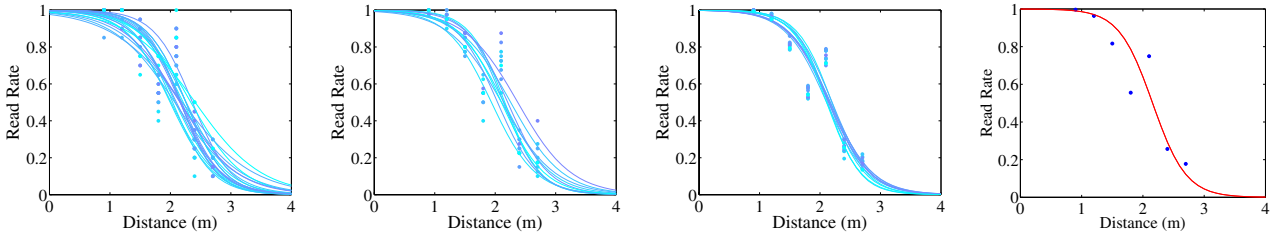
(b) Experimental Setting on the Shelf

Figure 1: Experimental Setting for Reader Detection Model



(a) 10 cycles (b) 20 cycles (c) 100 cycles (d) Average

Figure 2: Read Rate vs. Tag Distance for the Ground-Level Setting



(a) 10 cycles (b) 20 cycles (c) 100 cycles (d) Average

Figure 3: Read Rate vs. Tag Distance for the Waist-Level Setting

have access to a set of pairs $\{(p_{i1}, \ell_{i1}), (p_{i2}, \ell_{i2}), \ldots, (p_{im}, \ell_{im})\}$ for each reader R_i.

Equation (1) implies that $\ln(\frac{1}{p_i} - 1) = a_i \ell_i + b_i$. Thus we can employ the least square method to estimate a_i and b_i, using the transformed set of pairs $\{(\ln(\frac{1}{p_{i1}} - 1),\ \ell_{i1}), (\ln(\frac{1}{p_{i2}} - 1),\ \ell_{i2}), \ldots, (\ln(\frac{1}{p_{im}} - 1),\ \ell_{im})\}$.

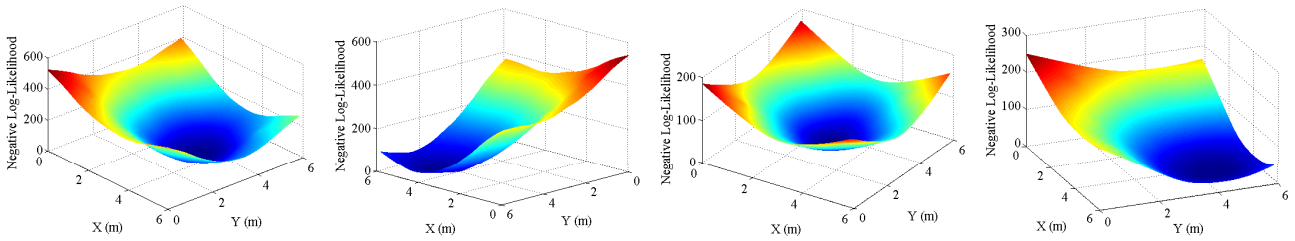
Our approach is cost-effective, since we can afford to distribute many cheap passive tags in the whole tracking region to enable the learning of an accurate model. Note that very far (or close) reference tags with read rate 0 (or 1) should be discarded, since Equation (1) does not allow $p = 0$ (or $p = 1$).

Our reader detection model is more accurate than the 3-state model in [3] which simply assumes a fixed constant read rate value among the *minor detection region*. More importantly, our model is adaptive to the environment and thus gives rise to more accurate location estimation.

3.2 Effect in Real Experimental Setting

We evaluated our reader detection model in the real indoor environments shown in Figures 1(a) and (b). In our experiments, 3 to 4 passive reference tags are placed in each of the four orthogonal directions around an antenna, with different distances to the antenna.

Figure 1(a) simulates the case where the readers/antennea and the reference tags are deployed on the ground, and each moving object sticks a tag at the bottom. For example, if the tracking objects are people, the tracking tags can be stick on the side of their shoes. To avoid the directional effect of the tags, each person can stick two tags of the same ID on both sides of their shoes. The reference tags are put at distances 1.05m, 1.35m, 1.65m and 1.95m from the antenna in two directions, and at distances 0.9m, 1.2m, 1.5m and 1.8m from the antenna in the other two directions. The tag positions are interleaved on the two pairs of directions, so as to provide more (*distance, read rate*) pairs for model fitting.



(a) Gound_level, Center (b) Gound_level, Border (c) Waist_level, Center (d) Waist_level, Border

Figure 4: Negative Log-Likelihood vs. Object Locations

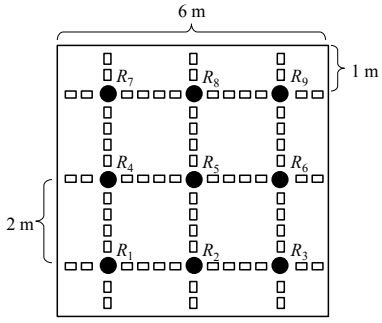


Figure 5: Tracking Area for Experimental Study

Figure 1(b) simulates the case where the readers/antennea and the reference tags are deployed 1m above the ground, e.g. stuck on the side of desks or walls. We make a shelf to build this setting. The shelf is made of styrofoam to minimize its impact on RFID detection. In this scenario, tracking tags can be stuck around the waists of people. The reference tags are put at distances 0.9m, 1.5m, 2.1m and 2.7m from the antenna in two directions, and at distances 1.2m, 1.8m and 2.4m from the antenna in the other two directions.

We use the Intermec IF61 smart reader which supports connection with 4 antennae, and connect it to an FCC-4 tuned dipole antenna produced by A.H. Systems. The antenna is omni-directional, which conforms to the models in the previous studies such as [3]. As for the passive tags, we use UHF Gen2 DogBone RFID tags from UPM Raflatac. We set the response timeout parameter so that it takes about 1 second to accomplish 5 interrogation cycles.

We measure the read rates based on 10 interrogation cycles for 20 times, 20 cycles for 15 times, and 100 cycles for 10 times. Figures 2 and 3 present the "read rate vs. distance" relations for the two experimental settings. For example, Figure 2(a) shows the tag read rates (points) and the learned curves for all the 20 runs based on 10 cycles in the ground-level setting. Figures 2(b)–(c) and 3(a)–(c) are similarly obtained. In Figures 2(d) and 3(d), the blue points are the average read rates of the 10 runs based on 100 cycles, and the red curves are obtained by least squares on these points.

It is obvious that the slopes of the curves in Figure 3 are gentler than those in Figure 2, which implies that read rate decays more slowly with distance in the waist-level setting, possibly due to less interference of the ground. Clearly, the read rate measurements based on 100 interrogation cycles are the most stable ones, and therefore, we determine the parameters a and b based on readings from the latest 100 cycles (around 20s). By fitting $p(\ell) = \frac{1}{1+e^{a\ell+b}}$ we have $a = 9.4831$ and $b = -16.2015$ in Figure 2(d), and $a = 3.6423$ and $b = -7.8597$ in Figure 3(d).

4. LOCATION INFERENCE MODEL

In this section, we describe our approach of estimating object locations using the dynamic reader detection model.

Consider the event that during the latest N interrogation cycles, the object with a tracking tag O responds to reader R_i for k_i times. This happens with probability $\binom{N}{k_i} p_i^{k_i}(1-p_i)^{N-k_i}$. If we assume that each reader detects the tag of each object independently, which is also assumed in other work such as [3], then the probability that O responds k_i times to reader R_i $(i = 1, 2, \cdots, n)$ is given by

$$L = \prod_{i=1}^{n} \binom{N}{k_i} p_i^{k_i}(1-p_i)^{N-k_i} \triangleq C \prod_{i=1}^{n} p_i^{k_i}(1-p_i)^{N-k_i}, \quad (3)$$

where $C = \prod_{i=1}^{n} \binom{N}{k_i}$ is a constant.

Our aim is to find the value of O's location $\vec{l} = (x, y)$ that maximizes the likelihood L, which is equivalent to minimizing the corresponding negative log-likelihood $\mathcal{L} = -\ln(L/C)$ (without considering the constant $C > 0$):

$$\mathcal{L} = -\sum_{i=1}^{n} [k_i \ln p_i + (N - k_i)\ln(1-p_i)] \triangleq -\sum_{i=1}^{n} \mathcal{L}_i \,. \quad (4)$$

Figure 5 shows a tracking area of 6 m $\times$ 6 m $=$ 36 m^2, where the 9 black circles represent readers/antennae, and the small rectangles represent reference tags. This setting will be used to evaluate the performance of PassTrack in Section 5.

Note that PassTrack actually allows for much more flexible deployment that caters to the actual building structures, as long as (1) for each antenna, there exist sufficient reference tags with different distances to it, and (2) for most of the indoor locations, a tracking tag there can be detected by at least 3 to 4 nearby antennae.

The first requirement ensures that there are sufficient learning samples for estimating the reader detection model parameters. The second requirement is called *detection range overlap*. Besides improving localization accuracy, it also allows the system to work even if some reader/antenna gets disconnected.

Refer to the tracking setting in Figure 5 again. We now denote the vector of response count information by $\vec{k} = \{k_1, k_2, \ldots, k_9\}$, where k_i denotes the response count obtained by reader R_i. Figure 4(a) demonstrates the values of the negative log-likelihood $\mathcal{L}$ at various locations when the actual location of the object to track is close to the center of the tracking area ($\vec{k} = \{0, 0, 0, 0, 0, 4, 0, 3, 5\}$), and Figure 4(b) is when the actual location is close to the border ($\vec{k} = \{0, 0, 0, 0, 5, 5, 0, 5, 0\}$). Both Figures 4(a) and 4(b) are for the ground-level setting. Figures 4(c) ($\vec{k} = \{0, 0, 0, 0, 2, 5, 0, 5, 5\}$) and 4(d) ($\vec{k} = \{0, 2, 0, 3, 5, 5, 1, 3, 0\}$) are for the waist-level setting, and are plotted in a similar manner. For Figures 4(a)–(d), the read rate estimation is based on 5 cycles (which takes around 1s).

We now study how to use the likelihood model of Equation (4) for efficient location inference. One approach is to impose a grid structure on the tracking area, and to evaluate the negative log-likelihood $\mathcal{L}$ using Equation (4) at each location on the grid. The location of the object to track is estimated as the location on the grid with the minimum value of $\mathcal{L}$. However, the granularity of the grid structure influences the accuracy of the estimation, and evaluation on a too fine-grained grid structure is time-consuming.

Another approach to find the location that minimize $\mathcal{L}$ is to use Newton's method [7], which is well-known for function optimization. Newton's method enables faster convergence than gradient descent, since it is based on the second-order Taylor approximation of the target function. As shown in Figures 4(a)–(d), $\mathcal{L}$ is almost convex, which is appropriate for applying Newton's method. Since $\mathcal{L}$ is not strictly convex, the choice of the initial location to start Newton's method is very important for avoiding being stuck at those non-convex locations, and we choose it to be the result location of the grid-based method described above, so as to make it close to the global minimum. In this case, a coarse-grained grid structure is adequate, and we adopt an 8×8 grid for the tracking area in Figure 5. Algorithm 1 shows the details of our Newton's method, which is adapted from [7].

Algorithm 1 Object Location Inference by Newton's Method

1: **given** tolerance $\epsilon > 0$
2: Compute the starting point (x_0, y_0) using grid search
3: $\vec{l} \leftarrow (x_0, y_0)$
4: **repeat**
5: Compute $\nabla\mathcal{L}(\vec{l})$
6: Compute $\nabla^2\mathcal{L}(\vec{l})$
7: Compute the Newton step: $\triangle\vec{l} \leftarrow -(\nabla^2\mathcal{L}(\vec{l}))^{-1}\nabla\mathcal{L}(\vec{l})$
8: Compute decrement: $\lambda^2 \leftarrow \nabla\mathcal{L}(\vec{l})^T(\nabla^2\mathcal{L}(\vec{l}))^{-1}\nabla\mathcal{L}(\vec{l})$
9: **quit** if $\lambda^2/2 \leq \epsilon$
10: Choose step size t by backtracking line search
11: $l \leftarrow \vec{l} + t\triangle\vec{l}$
12: **end repeat**

Line 10 of Algorithm 1 uses the backtracking line search algorithm shown in Algorithm 2. The default parameters we use for backtracking line search are $\alpha = 0.2, \beta = 0.6$ and $\epsilon = 10^{-6}$.

Algorithm 2 Backtracking Line Search

1: **given** a descent direction $\triangle\vec{l}$ at $\vec{l}$,
constants $\alpha \in (0, 0.5), \beta \in (0, 1)$
2: $t \leftarrow 1$
3: **while** $\mathcal{L}(\vec{l} + t\triangle\vec{l}) > \mathcal{L}(\vec{l}) + \alpha t(\nabla\mathcal{L}(\vec{l}))^T\triangle\vec{l}$ **do**
4: $t \leftarrow \beta t$
5: **end while**

In our experiments, we find that Algorithm 1 always terminates within 10 iterations. However, we still set an upperbound of 100 iterations for Algorithm 1. Another special case we handle is when the inferred location leaves the tracking area. In this case, the algorithm immediately returns the location estimation of the previous iteration. A third possible scenario is when an object is not detected by any reader/antenna within the current time window, in which case we use the old location estimation as the current one. We never encounter such a scenario in our experiments, although it is possible for the RFID readers/antennea to miss readings when there are too many tags in their detection ranges. We do not include the above details in the pseudo-code in Algorithm 1 for better readability.

Note that Algorithm 1 requires the computation of the gradient (see Line 5) and Hessian matrix (see Line 6) of $\mathcal{L}$. Fortunately, they have elegant forms that can be evaluated efficiently, as is formalized by Theorem 1 below.

THEOREM 1. *Given the negative log-likelihood function $\mathcal{L}$ (Equation (4)), its gradient $\nabla\mathcal{L}(\vec{l})$ and Hessian matrix $\nabla^2\mathcal{L}(\vec{l})$ are given by:*

$$\nabla\mathcal{L} = \begin{bmatrix} \frac{\partial\mathcal{L}}{\partial x} \\ \frac{\partial\mathcal{L}}{\partial y} \end{bmatrix} = \sum_{i=1}^{n} B_i \begin{bmatrix} \Delta x_i/\ell_i \\ \Delta y_i/\ell_i \end{bmatrix}, \tag{5}$$

$$\begin{aligned} \nabla^2\mathcal{L} &= \begin{bmatrix} \frac{\partial^2\mathcal{L}}{\partial x^2} & \frac{\partial^2\mathcal{L}}{\partial x\partial y} \\ \frac{\partial^2\mathcal{L}}{\partial y\partial x} & \frac{\partial^2\mathcal{L}}{\partial y^2} \end{bmatrix} \\ &= \sum_{i=1}^{n} \left(\frac{A_i}{\ell_i^2} \begin{bmatrix} \Delta x_i^2 & \Delta x_i\Delta y_i \\ \Delta x_i\Delta y_i & \Delta y_i^2 \end{bmatrix} \right. \\ &\quad \left. + \frac{B_i}{\ell_i^3} \begin{bmatrix} \Delta y_i^2 & -\Delta x_i\Delta y_i \\ -\Delta x_i\Delta y_i & \Delta x_i^2 \end{bmatrix} \right), \end{aligned} \tag{6}$$

where

$$\Delta x_i = x - x_i\,, \tag{7}$$

$$\Delta y_i = y - y_i\,, \tag{8}$$

$$A_i = a_i^2 N p_i(1 - p_i)\,, \tag{9}$$

$$B_i = a_i(k_i - N p_i)\,. \tag{10}$$

PROOF. Since x and y are symmetric, we only need to compute $\frac{\partial\mathcal{L}}{\partial x}$, $\frac{\partial^2\mathcal{L}}{\partial x^2}$ and $\frac{\partial^2\mathcal{L}}{\partial x\partial y}$. Note that $\mathcal{L}$ is a function of p_i (Equation (4)), which is in turn a function of l_i (Equation (1)), which is further a function of x and y (Equation (2)). For ease of presentation, we define the following notation:

$$\delta_i = \Delta y_i/\Delta x_i\,. \tag{11}$$

We will use the following property in the computation, which can be easily verified:

LEMMA 1. *The derivative of $p(\ell) = \frac{1}{1+e^{a\ell+b}}$ $(a > 0, b < 0)$ is $p'(\ell) = p(\ell)\cdot(p(\ell) - 1)$.*

By taking the partial derivative of $\mathcal{L}$ with respect to x, we have

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial x} &= -\sum_{i=1}^{n}\left[\frac{\partial\mathcal{L}_i}{\partial p_i}\cdot\frac{\partial p_i}{\partial(a\cdot\ell_i + b)}\cdot\frac{\partial(a\cdot\ell_i + b)}{\partial x}\right] \\ &= -\sum_{i=1}^{n}\left[\left(\frac{k_i}{p_i} + \frac{N - k_i}{p_i - 1}\right)\cdot p_i(p_i - 1)\right. \\ &\quad \left.\cdot\frac{a_i(x - x_i)}{\sqrt{(x - x_i)^2 + (y - y_i)^2}}\right] \\ &= -\sum_{i=1}^{n}\left[(N p_i - k_i)\cdot\frac{a_i(x - x_i)}{\sqrt{(x - x_i)^2 + (y - y_i)^2}}\right] \\ &= \sum_{i=1}^{n}\frac{a_i(k_i - N p_i)}{\sqrt{1 + \delta_i^2}} \\ &= \sum_{i=1}^{n} a_i(k_i - N p_i)\frac{\Delta x_i}{\ell_i}\,. \end{aligned} \tag{12}$$

Next, we compute the second derivatives:

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}}{\partial x^2} &= \frac{\partial}{\partial x}\left[\sum_{i=1}^{n} \frac{a_i(k_i - Np_i)}{\sqrt{1+\delta_i^2}}\right] \\
&= \sum_{i=1}^{n}\left[\frac{1}{\sqrt{1+\delta_i^2}} \cdot \frac{\partial}{\partial x}(a_i(k_i - Np_i))\right. \\
&\quad \left. + a_i(k_i - Np_i) \cdot \frac{\partial}{\partial x}\left(\frac{1}{\sqrt{1+\delta_i^2}}\right)\right] \\
&= \sum_{i=1}^{n}\left[\frac{-a_i N}{\sqrt{1+\delta_i^2}} \cdot \frac{\partial p_i}{\partial x}\right. \\
&\quad \left. - a_i(k_i - Np_i) \cdot \left(\delta_i(1+\delta_i^2)^{-\frac{3}{2}}\right) \cdot \frac{\partial \delta_i}{\partial x}\right] \\
&= \sum_{i=1}^{n}\left[\frac{-a_i N}{\sqrt{1+\delta_i^2}} \cdot \frac{a_i p_i(p_i-1)}{\sqrt{1+\delta_i^2}} - a_i(k_i - Np_i)\right. \\
&\quad \left. \cdot \left(\frac{\Delta y_i}{\Delta x_i}\left(\frac{\Delta x_i^2 + \Delta y_i^2}{\Delta x_i^2}\right)^{-\frac{3}{2}}\right) \cdot \left(-\frac{\Delta y_i}{\Delta x_i^2}\right)\right] \\
&= \sum_{i=1}^{n}\left[\frac{a_i^2 N p_i(1-p_i)}{1+\delta_i^2} + a_i(k_i - Np_i)\frac{\Delta y_i^2}{\ell_i^3}\right] \\
&= \sum_{i=1}^{n}\left[a_i^2 N p_i(1-p_i)\frac{\Delta x_i^2}{\ell_i^2} + a_i(k_i - Np_i)\frac{\Delta y_i^2}{\ell_i^3}\right].
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}}{\partial x \partial y} &= \frac{\partial}{\partial y}\left[\sum_{i=1}^{n} \frac{a_i(k_i - Np_i)}{\sqrt{1+\delta_i^2}}\right] \\
&= \sum_{i=1}^{n}\left[\frac{1}{\sqrt{1+\delta_i^2}} \cdot \frac{\partial}{\partial y}(a_i(k_i - Np_i))\right. \\
&\quad \left. + a_i(k_i - Np_i) \cdot \frac{\partial}{\partial y}\left(\frac{1}{\sqrt{1+\delta_i^2}}\right)\right] \\
&= \sum_{i=1}^{n}\left[\frac{-a_i N}{\sqrt{1+\delta_i^2}} \cdot \frac{\partial p_i}{\partial y}\right. \\
&\quad \left. - a_i(k_i - Np_i) \cdot \left(\delta_i(1+\delta_i^2)^{-\frac{3}{2}}\right) \cdot \frac{\partial \delta_i}{\partial y}\right] \\
&= \sum_{i=1}^{n}\left[\frac{-a_i N}{\sqrt{1+\delta_i^2}} \cdot \frac{a_i p_i(p_i-1)}{\sqrt{1+\delta_i'^2}} - a_i(k_i - Np_i)\right. \\
&\quad \left. \cdot \left(\frac{\Delta y_i}{\Delta x_i}\left(\frac{\Delta x_i^2 + \Delta y_i^2}{\Delta x_i^2}\right)^{-\frac{3}{2}}\right) \cdot \frac{1}{\Delta x_i}\right] \\
&= \sum_{i=1}^{n}\left[a_i^2 N p_i(1-p_i)\frac{\Delta x_i \Delta y_i}{\ell_i^2}\right. \\
&\quad \left. - a_i(k_i - Np_i)\frac{\Delta x_i \Delta y_i}{\ell_i^3}\right].
\end{aligned}$$

□

To compare with the likelihood-based method described above, we also incorporate a baseline algorithm based on the idea of nearest neighbors (NN), which are popular with the active-tag-based RFID localization systems (see Section 6). The location of an object O is estimated as $(x, y) = \sum_{i=1}^{n} w_i(x_i, y_i)$, where the weight w_j of reader R_j is empirically set to be $w_j = \frac{k_j}{\sum_{i=1}^{n} k_i}$ or $w_j = \frac{k_j^2}{\sum_{i=1}^{n} k_i^2}$, to give higher weight to a reader that detects O with a higher read rate. Our extensive experiments described in Section 5 have shown that the naive NN-based approach incurs several times more error than our Newton's-method-based approach, which verifies that, our Newton's-method-based approach best brings out the information contained in read rates.

As there are only 5 interrogation cycles in each second, if we estimate the location of an object for each second solely using the read rates that are measured based on these 5 cycles, we cannot obtain read rate values of high precision. Therefore, we use a sliding window of multiple cycles to measure the read rates. While a larger window usually improves the precision of read rate measurements for static objects, this is not the case for moving objects due to the outdated readings in the window. We will study the relationship between window size and localization accuracy in Section 5.5.

5. EXPERIMENTS

In this section, we evaluate the performance of PassTrack in the 6m×6m tracking area shown in Figure 5, which is equipped with 9 antennae. To save our research time, we simply use synthetic data instead of real data for the experiments. In order to run experiments in a wide variety of settings, we built a data generator that simulates the multi-antennae scenario according to the real-world study on one-antenna described in Section 3.2.

Data Generator. In each time step (which is 1s), our data generator first generates the locations of the moving objects to track, and then, the readings of each reader are generated according to its reader detection model: in each of the 5 cycles, the tag of a moving object is detected by the reader with the probability p computed from Equation (1).

To avoid generating too idealized readings, we design our data generator to also be able to generate noisy readings that better simulate the real-world scenarios. In the ideal case, the response count value conforms to the binomial distribution with standard deviation $\sqrt{Np(1-p)}$. Our data generator generates noisy readings by adding or subtracting (with equal probability) the original response count by $\tilde{\beta}\sqrt{Np(1-p)}$, where the random variable $\tilde{\beta} \sim$ *Uniform*$[0, \beta]$, and β is a user-specified parameter that controls the noise level. If the resulting response count is below 0 (or above N), it is set to 0 (or N).

We use $[w_\ell, w_u]$ to configure the sliding window to include the previous w_ℓ time steps and the succeeding w_u time steps besides the current one. For example, $[0, 0]$ denotes that the window contains only the 5 cycles in the current second, while $[-1, 1]$ denotes that the window contains the cycles from the previous second to the next second ($N = 15$).

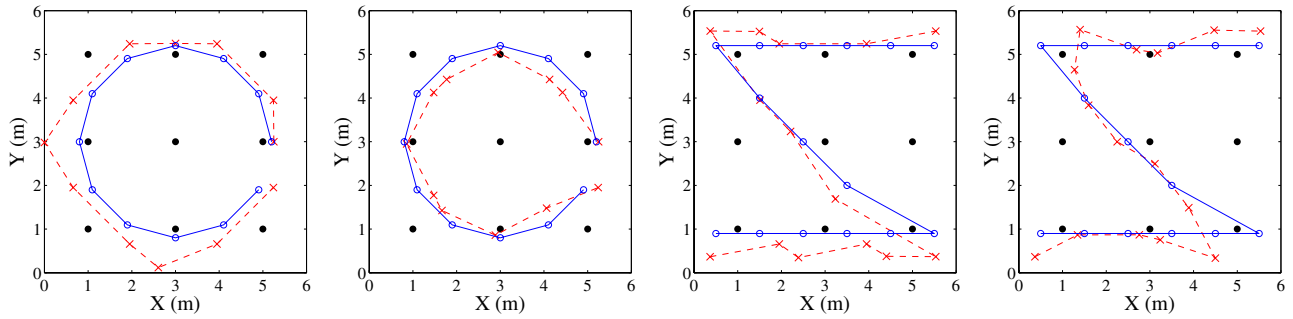
Unless otherwise stated, we fix the reader model parameters a_i and b_i during data generation and location inference. We set $a_i = 9.4831$ and $b_i = -16.2015$ ($a_i = 3.6423$ and $b_i = -7.8597$) for the experiments in the ground-level (waist-level) setting.

In the sequel, we denote our grid-based localization method as *Grid*, our Newton's method that uses *Grid* for initialization as *Newton*, our two nearest-neighbor-based methods as *NN* (where $w_j = k_j / \sum_{i=1}^{n} k_i$) and *NN2* (where $w_j = k_j^2 / \sum_{i=1}^{n} k_i^2$). We assume the speed of a moving object to be 1m/s to simulate walking speed.

All the experiments are run on a Lenovo IBM ThinkPad X201i laptop with a 2.53Hz Intel Core i3 GPU and 2GB memory.

5.1 Localization Accuracy for Moving Objects

To visualize the effectiveness of our localization approach in scenarios where objects are always moving, we generate the data that correspond to an object moving in circular and Z-shaped trajecto-



(a) Circle, Window $= [0, 0]$ (b) Circle, Window $= [-1, 1]$ (c) Z-Curve, Window $= [0, 0]$ (d) Z-Curve, Window $= [-1, 1]$

Figure 6: Real and Inferred Trajectories in the Waist-Level Setting

ries, and invoke *Newton* to estimate its location at each time step. The experiments are done in the waist-level setting and the results are shown in Figure 6. In Figures 6(a)–(d), the blue points (connected by the solid lines) are the true locations (trajectories) of the object, and the red points (connected by the dotted lines) are the inferred locations (trajectories). The estimated trajectory in Figure 6(b) has a smaller radius than the true circular trajectory, because the window contains the readings from the previous and the next second. For the experiments in Figures 6(a)–(d), *Newton* is able to recover the original trajectory of an object with reasonable accuracy: the average error never exceeds 30cm.

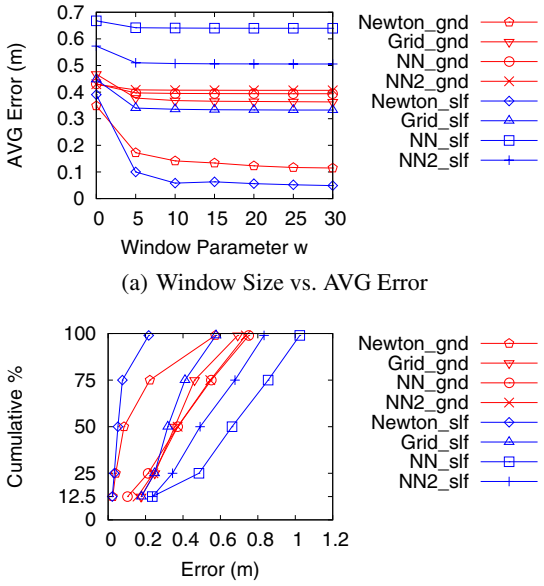
5.2 Localization Accuracy for Static Objects

A larger sliding window usually improves the precision of read rate measurements for static objects, which in turn tends to increase the accuracy of object location estimation. In this subsection, we study the relationship between window size and localization quality for static objects. We randomly generate the locations for 5 static objects, and run our localization algorithms on the data for 3000 time steps in both the ground-level setting and the waist-level setting, with different window sizes. We adopt the window configuration $[-w, w]$, where the parameter w controls the size of the window (which is $2w + 1$).

For each of the 15,000 (5×3000) location estimations, we compute its error distance as the Euclidean distance between the estimated location and the true location. One measurement of localization accuracy that we use is the average of these 15,000 error distances. Another measurement is the cumulative percentile of error distance which is also adopted in [1].

Figure 7(a) shows the average error distance of our localization algorithms with different window sizes, where the suffix "_slf" (or "_gnd") means that the corresponding algorithm works at the waist-level (or ground-level) setting. We can see that *Newton* incurs the least localization error and thus achieves the best accuracy. The average errors of all the algorithms decrease with the increment of window size, and when $w = 5$, *Newton_slf* is already around 10cm and *Newton_gnd* is already around 17cm. In this case, the window size is $2w + 1 = 11$, and as there are 5 cycles in each time step, only 55 interrogation cycles are used to estimate the read rates. This result is much better than the work of [10], which uses 100 cycles to estimate the read rates and still incurs an average error of 19cm.

Since the reader detection range is longer at the waist-level setting than at the ground-level one, an object tends to be detected by more readers. As a result, at the waist-level setting, *Newton* can use more information for location estimation and therefore incurs less error than if it works at ground-level, as illustrated in Figure 7(a). However, this is not the case for the NN-based algorithms: the



(a) Window Size vs. AVG Error

(b) Cumulative Percentile of Error When $w = 10$

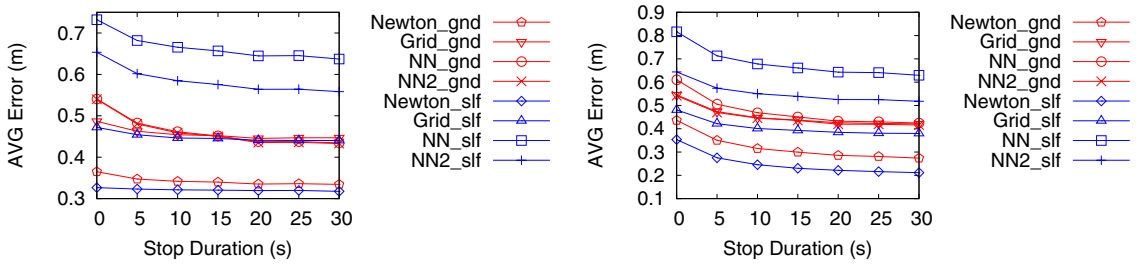
Figure 7: Effect of Window Size for Static Object Localization

waist-level versions are shown to be worse than the ground-level ones in Figure 7(a). This is because the NN-based heuristics lack the support of a formal underlying inference model.

Figure 7(b) shows the cumulative percentile of error distance for our localization algorithms when $w = 10$, from which we have the similar observations that *Newton* incurs the least error and the waist-level version works better than the ground-level one. From Figure 7(b), we can see that *Newton* can estimate the location of an object within 22cm error with 75% probability.

5.3 Effect of Object Moving Frequency

We studied the performance of our localization algorithms for constantly moving objects in Section 5.1 and for static objects in Section 5.2. However, a more practical scenario is when objects frequently change their locations, and stop for some time between two consecutive movements. For example, in inventory monitoring, the goods are not likely to change their locations all the time, although they may be moved to different shelves from time to time. The application of resident monitoring also requires to recognize activities such as "reading paper" (e.g. on sofa) and "using a computer" (e.g. on a desk), which involve the stay at a specific location for a while.



(a) Window $= [0, 0]$ (b) Window $= [-1, 1]$

Figure 8: Effect of Object Moving Frequency on Localization Accuracy

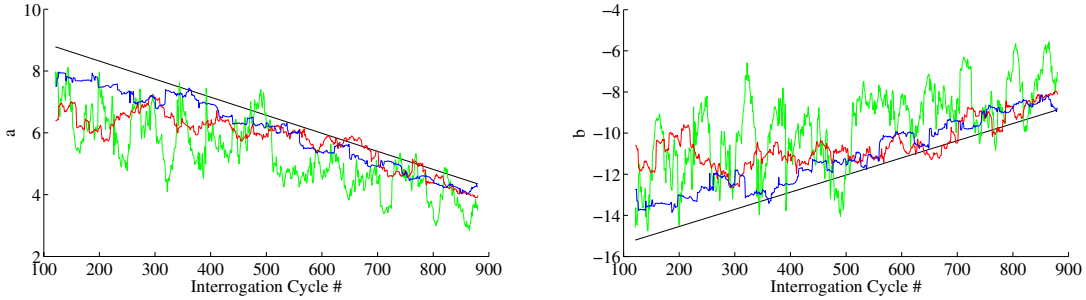


Figure 9: Learned v.s. True Model Parameters

To study the effect of object moving frequency on localization accuracy, we generate the trajectories of 5 objects that interleave between staying and moving for 3000 time steps. We parameterize our data generator with stop duration T, so that each object moves for k steps (turns around if it reaches the wall), where $k \in \{1, 2, 3, 4, 5\}$ is randomly picked, then stops for T time steps, and then repeats the above process until 3000 time steps are reached.

Figure 8(a) (or Figure 8(b)) shows the average error distance of our localization algorithms with window $[0, 0]$ (or $[-1, 1]$) for various stop durations. While both Figures 8(a) and (b) verify the intuition that all the algorithms incur less error for longer stop duration (and thus less frequent movements), the *Newton* algorithms have more dramatic drop in error with the increment of stop duration in Figure 8(b). This is because more interrogation cycles are used for read rate estimation when window $[-1, 1]$ is adopted, which leads to more accurate read rate estimation, and thus more accurate localization. Both figures confirm that *Newton* incurs the least error and the waist-level version works better than the ground-level one.

5.4 Results of Model Parameter Learning

In Section 3.2, we studied the quality of the reader detection models learned using read rate estimations of reference tags based on 10, 20 and 100 interrogation cycles, and showed that estimating read rates based on 100 cycles gives the most stable model.

In this subsection, we further study the effect of window size for reference tag read rate estimation on localization accuracy. While a larger window size gives more accurate read rate estimations of the reference tags, the model learned from these read rate estimations may not be close to the true reader detection model. This is because the true reader detection model changes with time, and the large window contains responses to early interrogation cycles. In fact, there is a delay between the learned model and the true one, and the larger the window size, the longer the delay.

To study the effect of window size on the difference between the learned model and the true one, we randomly generate the locations

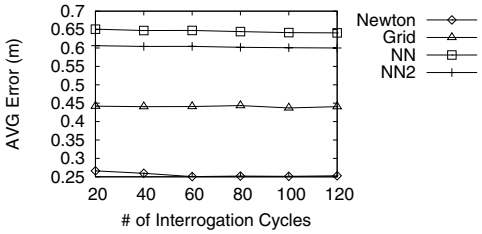
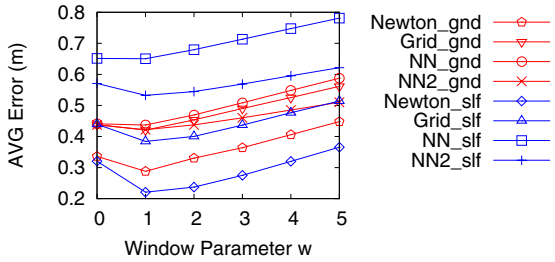


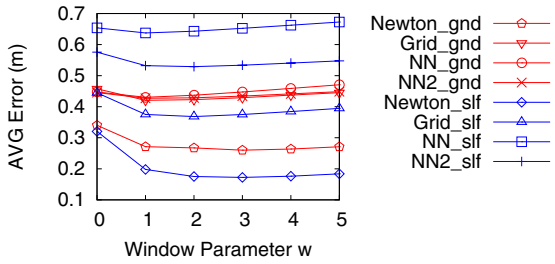
Figure 10: Window Size vs. Localization Accuracy

for 5 static objects, and run our localization algorithms on the data for 200 time steps (1000 cycles). We simulate the change of true reader detection model, by changing the model parameters a and b linearly from the ground-level parameters (at the beginning of the 1000 cycles) to the waist-level parameters (at the end of the 1000 cycles). Note that this is a more abrupt model change than what is likely to happen in a real world scenario. Then, we learn the reader detection model from read rate estimations of reference tags based on 20, 40, 60, 80, 100 and 120 interrogation cycles.

Figures 9(a) and (b) show both the true model parameters and the learned model parameters in the 200 time steps, where the black lines correspond to the true model parameters, and the green, red and blue curves correspond to the model parameters learned from read rate estimations based on 20, 60 and 120 interrogation cycles, respectively. In Figures 9(a) and (b), the learned model parameters present an obvious delay from the true parameters (i.e. the learned parameter curves are approximately the true parameter curves shifted towards the left), and the larger the number of interrogation cycles, the longer the delay (note that the colors of the curves from left to right are green, red, blue and black in order). Besides, model parameters learned from read rate estimations based on small windows such as the 20-cycle one (the green curve) are very unstable.



(a) Stop Duration = 20



(b) Stop Duration = 80

Figure 11: Effect of Window Size on Localization Accuracy

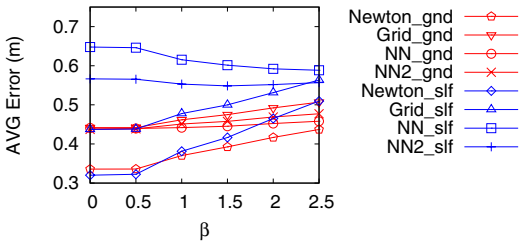
Figure 10 shows the average error distance of our localization algorithms with different window size for reference tag read rate estimation, where we use window $[-1, 1]$ for tracking tag read rate estimation. We can see that the 60-cycle window leads to the least localization error (around 25cm) for *Newton*. In the real scenario, the model change is not likely to be as abrupt as our experimental setting, and larger window size such as 100 cycles is likely to be a better choice.

5.5 Effect of Window Size for Moving Objects

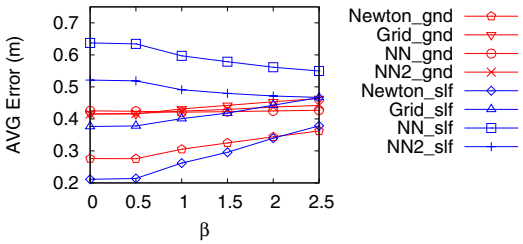
In Section 5.2, we have seen that a larger window increases the accuracy of location estimation for static objects. However, larger window size may not always imply higher localization accuracy for moving objects. On the one hand, a larger window contains more interrogation cycles, which give rise to read rate measurements of higher precision. On the other hand, the outdated readings in a larger window have a negative impact on the localization accuracy, since the tracking object is assumed to be at a fixed location throughout the time in the window.

Therefore, with the increment of window size, one can expect that the localization error drops when the window size is small, and then rises as the window size further increases. The optimal window size depends on the average moving frequency of the tracking objects, and needs to be tuned according to the application. For example, while window $[-1, 1]$ is better than window $[0, 0]$ for tracking constantly moving objects as shown in Section 5.1, the reverse is true for the experiments on static objects in Section 5.2.

To study the relationship between optimal window size and object moving frequency, we generate the trajectories of 5 objects for 3000 time steps with different stop durations T. Part of the results are shown in Figure 11. Figure 11(a) shows the average error distance of our localization algorithms when $T = 20$, where we can see that the optimal window parameter is $w = 1$. In Figure 11(b) which corresponds to $T = 80$, we can see that the optimal window parameter is $w = 3$. To sum up, the optimal window size is larger for objects that move less frequently.



(a) Window = $[0, 0]$



(b) Window = $[-1, 1]$

Figure 12: Effect of External Interference

5.6 Tolerance to External Interferences

So far, we have been assuming that the response count of a tag to a reader within N interrogation cycles strictly follows the binomial distribution. In this set of experiments, we study the tolerance of our algorithm to external interference by varying the parameter β of our data generator, which is described at the beginning of this section. We set the maximum value of β to be 2.5 so that the read count can be biased by at most 2.5 times its standard deviation, which is a very significant deviation.

Figure 12(a) (or Figure 12(b)) shows the average error distance of our localization algorithms with window $[0, 0]$ (or $[-1, 1]$) for various noise levels, where we can see that the error of *Newton_slf* increases more dramatically than and finally exceeds that of *Newton_gnd*, and therefore *Newton_gnd* is more tolerant to external interference than *Newton_slf*. From the figures we can see that our localization algorithms, including *Newton*, all have good tolerance to external interference, and *Newton* still incurs the least error in noisy environments.

5.7 Scalability

To test the scalability of our localization algorithms, we generate the trajectories of 20 constantly moving objects for varying time steps t, and run our algorithm on these data with window set to $[-1, 1]$. The execution times of all the algorithms increase linearly with the total number of location estimations (which is $20t$).

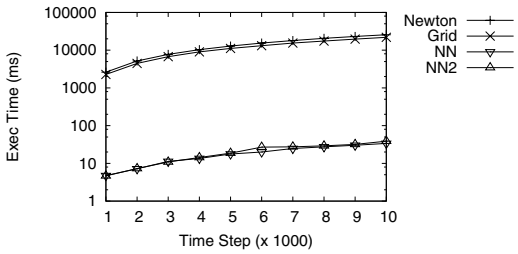


Figure 13: Scalability Results

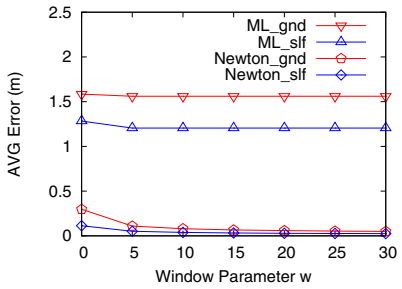


Figure 14: *Newton* vs. *Multilateralization* (*ML*)

Figure 13 shows the results of our scalability test, where the NN algorithms take only around 0.1% of the time consumed by *Newton* on the same number of location estimations, and *Grid* takes around 85% of the time consumed by *Newton*. For $t = 10{,}000$, the 200,000 ($= 20t$) location estimations are accomplished by *Newton* in less than 26s. Therefore, over 7500 location estimations can be done for each second by an ordinary computer, which is sufficient for supporting large scale real-time location tracking applications.

5.8 Comparison with Multilateration

In this subsection, we compare our best algorithm, i.e. *Newton*, with the best algorithm, i.e. *Multilateralization* (*ML*), in the pioneering work of [10] that uses passive tags for indoor localization.

We randomly generate the locations for 4 static objects, and run both localization algorithms on the data for 3000 time steps in both the ground-level setting and the waist-level setting, with different window sizes. Figure 14 shows the results of comparison, where we can see that the average error of *ML* is over 1m. In fact, *ML* is not as accurate as the nearest neighbor based methods, let alone *Newton*. The low accuracy of *ML* is caused by the small number (3 to 4) of readers that can detect an tracking object, since *ML* is effective only when many readers can detect an object. However, for passive tags with low detection range, this implies a dense deployment of readers/antennea, which is too costly to be carried out.

6. RELATED WORK

Most existing localization systems are based on the active RFID technology[1, 9, 11, 12, 14]. The nearest neighbor based algorithms are adopted by RADAR [9] and LANDMARC [1] for location inference, where the location of the target is estimated by properly averaging the locations of the reference points whose received signal strength distribution is similar to that of the target. Other more advanced models based on the signal propagation characteristics include those proposed in [11, 12, 13, 14], such as multilateration and Bayesian graphical modeling.

Passive RFID technology has been studied for data cleaning purpose [2, 3, 5, 6, 8]. However, very few work is done to track objects using passive tags, except the pioneering work of [10], which is still primitive in that its experiments are conducted in a very small area of 1.83m×1.83m.

7. CONCLUSION

Motivated by many advantages of passive RFID tags such as low tag cost and convenience of maintenance, we explore the techniques of using passive tags for indoor RFID localization. Our localization system, PassTrack, learns for each reader an adaptive reader detection model that has good tolerance to external interference. Several algorithms are supported by PassTrack for location inference based on read rates of passive RFID tags, and the best one, *Newton*, incurs an average localization error of around 30 cm, and is able to carry out over 7500 location estimations per second on an ordinary machine. The excellent tracking accuracy and efficiency of PassTrack enables it to support large-scale real-time indoor localization applications.

8. ACKNOWLEDGEMENTS

This work is partially supported by HKUST RFID Center under grant numbers ITP/022/02LP and SSRI08RGC, and GRF under grant number HKUST 617610.

9. REFERENCES

[1] L. M. Ni, Y. Liu, Y. C. Lau and A. P. Patil. “LANDMARC: Indoor Location Sensing Using Active RFID”. In *Wireless Networks*, 2004.

[2] S. R. Jeffery, M. Garofalakis and M. J. Franklin. “Adaptive Cleaning for RFID Data Streams”. In *VLDB*, 2006.

[3] H. Chen, W. Ku, H. Wang and M. Sun. “Leveraging Spatio-Temporal Redundancy for RFID Data Cleansing”. In *SIGMOD*, 2010.

[4] S. Han, J. Kim, C.-H. Park, H.-C. Yoon and J. Heo. “Optimal Detection Range of RFID Tag for RFID-Based Positioning System Using the k-NN Algorithm”. In *Sensors*, 2009.

[5] M. Kodialam and T. Nandagopal. “Fast and Reliable Estimation Schemes in RFID Systems”. In *Mobicom*, 2006.

[6] T. Tran, C. Sutton, R. Cocci, Y. Nie, Y. Diao and P. Shenoy. “Probabilistic Inference over RFID Streams in Mobile Environments”. In *ICDE*, 2009.

[7] S. Boyd and L. Vandenberghe. “Convex Optimization”. *Cambridge University Press*. http://www.stanford.edu/∼boyd/cvxbook/bv_cvxbook.pdf

[8] L. W. F. Chaves, E. Buchmann and K. Böhm. “TagMark: Reliable Estimations of RFID Tags for Business Processes”. In *KDD*, 2008.

[9] P. Bahl and V. N. Padmanabhan. “RADAR: An In-Building RF-based User Location and Tracking System”. In *INFOCOM*, 2000.

[10] J. Zhou and J. Shi. “RFID Localization Algorithms and Applications — A Review”. In *Journal of Intelligent Manufacturing*, Vol. 20, No. 6, pp. 695–707, 2009.

[11] A. Savvides, C. Han and M. B. Srivastava. “Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors”. In *MOBICOM*, 2001.

[12] D. Madigan, E. Einahrawy, R. P. Martin, W. Ju, P. Krishnan and A. S. Krishnakumar. “Bayesian Indoor Positioning Systems”. In *INFOCOM*, 2005.

[13] T. Roos, P. Myllymäki and H. Tirri. “A Statistical Modeling Approach to Location Estimation”. In *IEEE Trans. Mob. Comput.* (TMC)1(1):59–69 (2002).

[14] R. Öktem and E. Aydin. “An RFID Based Indoor Tracking Method for Navigating Visually Impaired People”. In *Turk J Elec Eng & Comp Sci*, Vol. 18, No. 2, 2010.

[15] A. Monreale, F. Pinelli, R. Trasarti and F. Giannotti. “WhereNext: A Location Predictor on Trajectory Pattern Mining”. In *KDD*, 2009.

[16] F. Giannotti, M. Nanni, F. Pinelli and D. Pedreschi. “Trajectory Pattern Mining”. In *KDD*, 2009.

[17] J. Lee, J. Han and K. Whang. “Trajectory Clustering: A Partition-and-Group Framework”. In *SIGMOD*, 2007.

[18] J. Lee, J. Han and X. Li. “Trajectory Outlier Detection: A Partition-and-Detect Framework”. In *ICDE*, 2008.