Enabling RFID-Based Tracking for Multi-Objects with Visual Aids: A Calibration-Free Solution

Chunhui Duan, Wenlei Shi, Fan Dang and Xuan Ding
School of Software and BNRist, Tsinghua University, China
Email: duanch@tsinghua.edu.cn, shiwenlei2012@gmail.com, {dangfan, dingxuan}@tsinghua.edu.cn

***Abstract*—Identification and tracking of multiple objects are essential in many applications. As a key enabler of automatic ID technology, RFID has got widespread adoption with item-level tagging in everyday life. However, restricted to the computation capability of passive RFID systems, locating or tracking tags has always been a challenging task. Meanwhile, as a fundamental problem in the field of computer vision, object tracking in images has progressed to a remarkable state especially with the rapid development of deep learning in the past few years. To enable lightweight tracking of a specific target, researchers try to complement computer vision to existing RFID architecture and achieves fine granularity. However, such solution requires calibration of the cameras extrinsic parameters at each new setup, which is not convenient for usage. In this work, we propose Tagview, a pervasive identifying and tracking system that can work in various settings without repetitive calibration efforts. It addresses the challenge by skillfully deploying the RFID antenna and video camera at the identical position and devising a multi-target recognition schema with only the image-level trajectory information. We have implemented Tagview with commercial RFID and camera devices and evaluated it extensively. Experimental results show that our method can archive high accuracy and robustness.**

***Index Terms*—Identification, tracking, RFID, computer vision**

I. Introduction

The ability to detect, track and identify multiple objects is crucial in many domains such as automated surveillance, goods monitoring, human-robot interaction, *etc*. A typical application lies in today's retail stores. The concept of checkout-free shopping has swept the retail market in the past few years, attracting both businesses and consumers with the frictionless shopping experience provided. To enable such intelligent service, one important technology is to identify and keep track of the goods concerned with high accuracy. Another potential usage is automated surveillance for security purpose. In the management of many warehouses and buildings, administrators want to know and monitor the movement of specific commodities or individuals. In all the above-mentioned tasks, both the ID and trajectory information of targets are valuable, which should be accurately acquired at the same time.

To realize identification and tracking for multiple objects, one possible solution is to utilize radio frequency identification (RFID) technology. As a key enabler of automatic ID technology, RFID offers an appealing alternative when compared against traditional barcodes, given the nature of non-line-of-sight (NLOS) communication and high reading rate of even multiple objects simultaneously. Many manufacturers today have already attached RFID tags to their products. But due to the limitation of passive tag's computation capability, it has always been a challenging task to localize or track tags, especially moving ones. State-of-the-art methods [1], [2] either require dedicated device (like USRP) or massive deployment costs (reference tags or antennas) to achieve high precision.

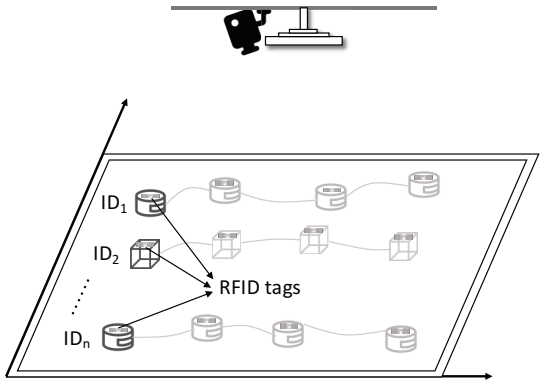


Fig. 1: Scene of Tagview. There are multiple moving objects carrying RFID tags. The video camera and RFID antenna are deployed in the same position.

As a fundamental problem within the field of computer vision (CV), remarkable breakthroughs have been made in object detection and tracking recently. Thanks to the proliferation of high-performance computers, the availability of high quality and inexpensive video cameras and the significant evolution of deep neural networks, it becomes feasible and affordable to keep track of even multiple moving objects in the image with high accuracy and low overhead. However, the most essential drawback of CV is that it can hardly identify specific targets among a set of moving objects, or in other words, it fails to distinguish one object from another.

To address the above issues, prior work [3] proposes a novel approach named TagVision, which supplements the RFID identification functionality with fine-grained tracking ability by combining computer vision technology. To be exact, one video camera and RFID antenna are deployed to obtain the trajectories of moving objects and phase sequence of the target tag. To reveal the location of the tag, TagVision tries to match it to one motion blob that is most likely to carry it. The rationale is that the measured phase of the tag should be consistent with the theoretical value computed with the true object's real-world trace information. Although TagVision offers a mean-

ingful technique for object identification and tracking task, the following two limitations remain to be overcome. First, TagVision mainly focuses on the identification and tracking of a single target, while a more complicated multi-object solution is actually desired in practical scenarios. Second, for tracking purpose, TagVision requires to calibrate the camera for its parameters (especially extrinsic parameters) in advance, so as to establish a complete transformation between the physical world coordinates and image pixels. Moreover, the calibration effort is not one-time-only, and need to be done at every new setup when the camera's posture changes or the world frame alters, which makes the system inflexible and inconvenient to use in practice.

Motivated by the above limitations, we propose Tagview, a calibration-free, lightweight, and fine-grained identifying and tracking system for multiple objects, which can work without troublesome camera calibration efforts. To achieve this, a video camera is deployed together with an RFID antenna, located at an identical position directly above the surveillance region, as shown in Fig. 1. The basic idea of Tagview is described as below. We first detect and track the moving objects in image-level utilizing state-of-the-art deep learning-based algorithms. Then by analyzing the geometric relationship of moving objects' image trajectories and RFID tags' phase sequences, we come up with a linear model to measure the *consistency* between a given pair of tag and object. After that, we design an innovative target recognition schema, where we formalize a weighted bipartite graph with the two sides representing the moving objects and tags respectively, and try to figure out a set of edges (*i.e.,* tag-object pairs) that maximize the sum of weights without violating certain constraints (more details will be given in Section V). Finally, by combining the identification and tracking results, we acquire the accurate identity and trajectory of each target.

Compared against previous work [3], Tagview has the following key advantages. First, Tagview is capable of identifying and tracking *multiple* moving objects without introducing extra hardware. Even in some exceptional circumstances, for example, objects move out of the surveillance region and later come back, we are still able to correctly recognize their identities. We believe such a feature would be useful in many scenarios. Second, our system offers a one-fit-all solution for object identification and tracking. By placing the camera and antenna in the same location and further devising a series of tricky algorithms, we free users from the troublesome camera calibration procedure, which is mandatory in TagVision. As we have no prior knowledge required on the relative position of the camera, Tagview can still work when the device moves or world system alters.

Contributions. In summary, this paper makes the following contributions:

- We propose a multi-object identification and tracking approach, which successfully combines the computer vision and RFID technologies with only a pair of the properly deployed RFID antenna and video camera. As a user-friendly system, Tagview need not require users to conduct camera calibration each time the setup changes.
- We present a novel identification schema that can operate with image-level trajectory information. A series of tricky algorithms are designed to overcome the negative impacts of imperfect measurements and other abnormal situations.
- We have implemented a prototype system for Tagview with commercial off-the-shelf (COTS) RFID and camera equipment, and evaluated it with extensive experiments. The final identification accuracy can reach as high as 0.98 on average, which demonstrates the practicality and effectiveness of our design.

Roadmap. The remainder of the paper is organized as follows. The main design of Tagview is overviewed in Section II. We introduce the object tracking mechanism in Section III. The technical details of our identification schema are elaborated in Section IV and Section V. We present the implementation and evaluation of our system in Section VI. We review related work in Section VII, and finally conclude this paper in Section VIII.

II. Overview

Tagview is an identification and tracking solution for multiple objects based on the combination of RFID and computer vision technologies. Fig. 1 gives an illustrative example of our system. In the scene, there are multiple moving objects (also referred to motion blobs), carrying RFID tags with various IDs. To relieve the camera calibration effort, we tactfully deploy an RFID antenna together with a video camera. The camera is mounted at the same location of the antenna on the ceiling, providing a bird's-eye view of the whole surveillance region. Specifically, Tagview decomposes the object recognition and tracking task into the following steps:

- With the image frames captured by the camera as input, Tagview goes through a multi-object detecting and tracking framework with the mechanism in Section III.
- Tagview analyzes the mathematical relationship between tags' phase measurements collected by an RFID reader, and image traces of moving objects acquired from the previous step. A linear relation is modeled to measure the consistency between these two modalities of data (see Section IV).
- Tagview realizes identification of targets by figuring out mapping between objects and tags utilizing the proposed algorithms in Section V.

The next few sections will elaborate on the above steps, providing the technical details.

III. Image-Level Object Tracking

Studies in object detection and tracking has flourished in recent years, and state-of-the-art trackers can achieve multi-object tracking with high accuracy and robustness. In consider of computation overhead and effectiveness, we choose a tracking-by-detection framework. More details will be presented in this section.

A. Multiple Object Detector

Given the image frame as input, the goal of object detection is to recognize instances of a predefined set of object classes (*e.g.,* humans, cars) and describe the locations of each detected object using a bounding box. Recently, deep learning techniques (typically based on convolutional neural network or CNN) have emerged with striking success in computer vision domain, serving as powerful means for learning feature representations automatically and directly from image data.

In this work, we adopt the single shot detector (SSD) [4] for object detection, which is a significant one-step framework based on global regression/classification, mapping straightly from image pixels to bounding box coordinates and class probabilities. The core idea of SSD is to discretize the output space of bounding boxes into a fixed set of default boxes over different aspect ratios and scales per feature map location. The network has multiple feature layers, with each produces a fixed set of detection predictions using a set of convolutional filters. For a feature layer of size $m \times n$ with p channels, the basic element for predicting parameters of a potential detection is a $3\times3\times p$ small kernel that produces either a score for a category or a shape offset relative to the default box coordinates. At each of the $m \times n$ locations where the kernel is applied, it produces an output value. The network is trained with a weighted sum of localization loss (*e.g.,* Smooth L1 [5]) and confidence loss (*e.g.,* Softmax).

B. Continuous Frame Tracker

In the tracking-by-detection strategy, a major task of multi-object tracking (MOT) is to associate object detections on a new video frame with previous ones to form trajectories of the targets. We leverage the simple online and realtime tracker (SORT) [6], which is an efficient and pragmatic online tracking approach. It approximates inter-frame displacements of each object with a linear constant velocity model which is independent of other objects and camera motion. The state of each target is modeled as:

$$[x_I, y_I, \alpha, \beta, \dot{x}_I, \dot{y}_I, \dot{\alpha}]^{\mathrm{T}}, \tag{1}$$

where x_I and y_I represent the horizontal and vertical pixel location of the target's center, while α and β represent the scale (area) and the aspect ratio of the targets bounding box. To assign new detections to existing targets, the algorithm computes a cost matrix as the intersection-over-union (IOU) distance between each detection and all predicted bounding boxes from the existing targets. When detection is associated with a target, the detected bounding box is used to update the target state where the velocity components are solved optimally via a Kalman filter framework [7].

C. Principle of Camera Imaging

The aforementioned tracking of objects is done in image-level. A 2D point in an image frame is a projection of a 3D point in the physical world, and their mathematical relationship is modeled by the camera parameters. Fig. 2 illustrates a simple pinhole camera model. The extrinsic parameters define the location and orientation of the camera with respect to the world coordinate system (X_W, Y_W, Z_W), so would change if the camera moves or the world frame alters. The intrinsic parameters (such as focal length, image center, and distortion) allow a mapping between camera coordinates and pixel coordinates (x_I, y_I) in the image frame, and they are internal and fixed to a particular camera/digitization setup. Many of today's commercial cameras can provide information on their intrinsic parameters. Even if not, a one-time calibration procedure with the technique in [8] would suit all devices of the same model.

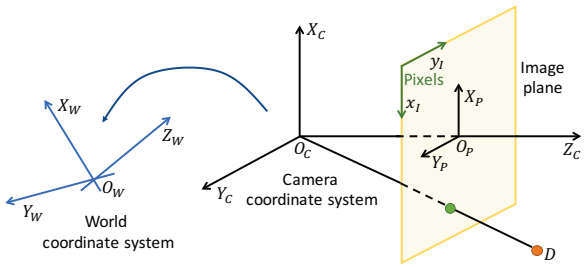


Fig. 2: Camera model. A 3D point is related to its 2D projection in the image frame through the camera's intrinsic and extrinsic parameters.

In this work, we presume that the camera intrinsic parameters are known in advance, which is not a harsh assumption and can be easily met in practice. Then given an original image, we first preprocess it by eliminating distortion and unifying aspect ratio to make the picture and real scene maintain a constant scaling factor. Different from prior work [3] which requires both the knowledge of camera's intrinsic and extrinsic parameters so as to establish a complete transformation from the real-world coordinates to image pixels, here Tagview only operates on the image-level and can still work when the camera position or world coordinates change. In many actual applications, what people really concern is not the absolute location or trace of an object, but the relative position of that object with regard to other ones. So it is reasonable that we focus on the image-level information of objects.

IV. Analyzing Tag-Object Relationship

In this section, Tagview tries to figure out the relationship between RFID tags' backscatter signals and the acquired image-level trajectories of moving objects.

A. Modeling Backscatter Signal

As Fig. 3 illustrates, the camera and RFID antenna are deployed in the same position A facing the surveillance plane, while A' is the projection of A on the physical plane with height $|AA'| = h$. When a tagged object B moves along a trace, suppose at an arbitrary time t, its distances to A and A' are $|BA| = d$ and $|BA'| = s$ respectively. In backscatter systems, the RF phase is a basic attribute of a wireless signal

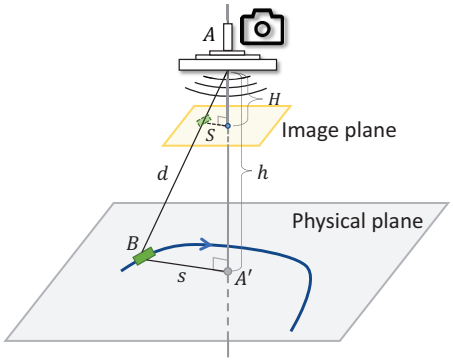
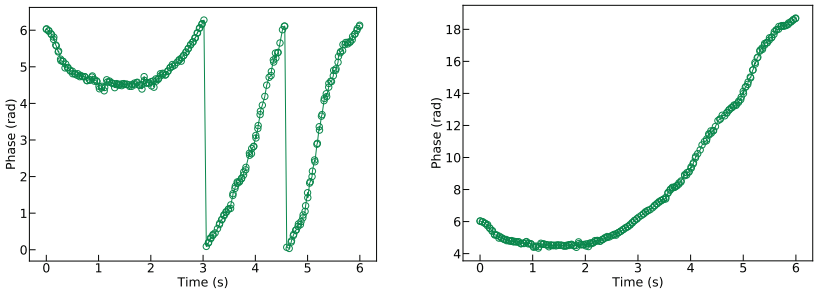


Fig. 3: Geometric relation of a moving object and an RFID antenna / camera



(a) Raw phase (b) Smoothed phase

Fig. 4: Phase measurements. (a) Raw phase sequence. (b) We smooth the phase by splicing adjacent split parts together.

and can be reported by commercial RFID readers [9]. Then the tag's phase shift [10] can be expressed as:

$$\begin{aligned}\theta(t) &= \left(\frac{2\pi}{\lambda} \times 2d(t) + c\right) \bmod 2\pi \\ &= \left(\frac{4\pi}{\lambda}\sqrt{h^2 + s^2(t)} + c\right) \bmod 2\pi,\end{aligned} \tag{2}$$

where λ is the wavelength, and the term c denotes a constant phase shift caused by the devices hardware characteristics. Note the total distance is $2d$ because the signal traverses a double distance back and forth in backscatter communication.

Fig. 4(a) gives an example of the measured phase sequence. It is split into many short discontinuous parts due to the $\bmod$ operation in Eqn. 2. For better analysis, we first smooth the curve by splicing adjacent split sub-sequences together. The smoothed phase is shown in Fig. 4(b).

As discussed in the previous section, for two points, the distance between them in the real physical world is proportional to that in the image pixel plane. Formally, the following expression holds:

$$\frac{s(t)}{S(t)} = \frac{h}{H}, \tag{3}$$

where S is the pixel value of s in the image, namely the distance in pixel from the image center O_I to the image point B_I of B. H is a constant factor representing the pixel distance from the principal point to the image plane, which can be calculated through the camera's intrinsic parameters. Substituting Eqn. 3 into Eqn. 2, we have

$$\theta(t) = \frac{4\pi h}{\lambda}\sqrt{1 + \frac{S^2(t)}{H^2}} + c. \tag{4}$$

Here we omit the $\bmod$ operation in Eqn. 2 as we have smoothed the raw phase. Record $\sqrt{1 + \frac{S^2(t)}{H^2}}$ as $\gamma(t)$. Since $S(t) = \sqrt{x_I^2 + y_I^2}$, γ can be directly computed with the object's image-level trajectory. For the sake of description, we call variable γ as **translating factor** in the rest parts of this paper. Apparently, θ is linearly dependent on parameter γ as shown below:

$$\theta(t) = a \times \gamma(t) + b. \tag{5}$$

B. Measuring Tag-Object Consistency

Since there are gaps between the frame rate (30 fps) of video camera and reading rate (about 50 Hz) of RFID readers. The collected samples of image traces and phase sequences may not align with each other. We first preprocess the two types of data with Hermite interpolation method to make them align in time domain. Then given a phase sequence $\Theta = \{(t_1, \theta[t_1]), (t_2, \theta[t_2]), \cdots\}$ ($\theta[t]$ is the acquired phase value at time t) and a translating factor sequence $\Gamma = \{(t_1, \gamma[t_1]), (t_2, \gamma[t_2]), \cdots\}$, if they are consistent with the same moving target, they should satisfy a linear relationship according to Eqn. 5. Therefore, we utilize linear regression to model the data and estimate the coefficients a and b. Mathematically, a *least squares* estimator is applied, which minimizes the sum of squared discrepancies between observed data and their expected values:

$$\arg\min_{a,b} \sum_i \varepsilon_i^2, \text{ where } \varepsilon_i = \theta[t_i] - (a\gamma_i + b). \tag{6}$$

Further, to evaluate the goodness-of-fit of the linear relation, we utilize the *coefficient of determination* (R-squared, denoted R^2), which measures the proportion of the variance in the dependent variable that is predictable from the independent variable:

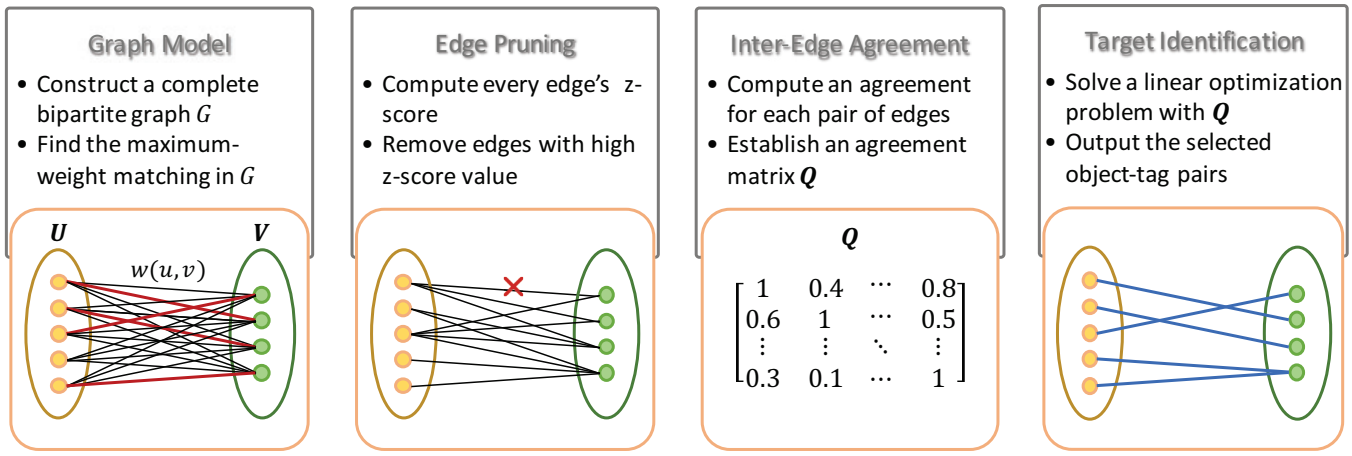
$$R^2 = 1 - \frac{\sum_i \varepsilon_i^2}{\sum_i \left(\theta[t_i] - \bar{\theta}\right)^2}, \tag{7}$$

where $\bar{\theta}$ denotes the mean of the observed phase data.

As we know, the linear relation in Eqn. 5 only holds well if the phase and trajectory are acquired from the same moving object. Thus, the value of R-squared can indicate the consistency between a given pair of tag and object to some extent. In other words, a higher R^2 means the tag is more likely to belong to that moving object because the corresponding phase and trajectory data fit better under linear regression.

V. Real-World Target Identification

In this section, we will elaborate on how to recognize the real-world identity of each moving object with only its image trace as input.



(a) Graph model (b) Edge pruning (c) Inter-edge agreement (d) Target identification

Fig. 5: Workflow of identification schema. (a) We first construct a complete graph with its two sides representing the moving objects and tags. (b) Then we prune invalid edges according to their z-score values. (c) We obtain an agreement matrix utilizing tag's RSSI. (d) We eventually solve an optimization problem to identify all the image traces.

A. Matching in Bipartite Graph

Suppose that we obtain m motion blobs' image trajectories and n tags' phase sequences. Note that here $m \geq n$ as one tag may corresponds to more than one traces when it moves out of the imaging range and later comes back. To identify and get the traces of all the tagged objects, we try to assign each image trajectory to a possible phase sequence.

To be specific, we establish a *complete bipartite graph* $\mathbb{G} = (\mathbb{U}, \mathbb{V}, \mathbb{E})$, where each vertex in $\mathbb{U}$ represents a mobile object with a known image trajectory and each one in $\mathbb{V}$ represents an RFID tag with a certain phase sequence. $\mathbb{U}$ and $\mathbb{V}$ are two disjoint and independent sets, where every vertex in $\mathbb{U}$ is connected to every vertex in $\mathbb{V}$, as shown in Fig. 5(a). For an arbitrary edge $e \in \mathbb{E}$ that connects a given pair of vertices (u, v) where $u \in \mathbb{U}, v \in \mathbb{V}$, we assign a *weight* $w(u, v)$ to e, which is set to the same value as the R-squared calculated through Eqn. 7.

We first try to find the maximum-weight matching in the bipartite graph $\mathbb{G}$, by exploiting the Hungarian matching algorithm, also called the Kuhn-Munkres algorithm [11].

B. Pruning Invalid Edges

Once the maximum-weight matching is found, we get a one-to-one mapping of motion blobs and RFID tags, which is most likely to be consistent with the ground truth. But such a mapping can not be directly used to identify targets because of the following two reasons. First, the mapping results are not entirely credible in certain cases. Consider two objects with translating factors $\gamma_1(t), \gamma_2(t)$ that satisfy $\gamma_1 = k\gamma_2$ (k is a constant), then they both would acquire the same goodness-of-fit with a given tag under linear regression (*i.e.,* their weights in the bipartite graph are very similar), which makes them easy to be falsely matched. Besides, due to imperfect measurements caused by surrounding noise in a practical indoor environment, the trajectory and phase data may not fit very well even if they are acquired from the same tag.

Recalling Eqn. 5, we come to the following key observation.

Observation V.1. *All correctly matched pairs of objects and tags should maintain the same coefficient* $a = \frac{4\pi h}{\lambda}$ *in the linear model.*

Given the maximum-weight matching $\mathbb{M}$ ($|\mathbb{M}| = n$), it is reasonable to assume that most of the edges in $\mathbb{M}$ agree with the ground truth. The detailed evaluation results is given in Section VI. Let $a_1, a_2, \cdots, a_n$ be the linear coefficients (computed through Eqn. 6) of all the connected object-tag pairs in $\mathbb{M}$, with their mean value $\bar{a}$ and variance σ^2. According to the aforementioned observation, we can use $\bar{a}$ as an estimate for the true linear term $\frac{4\pi h}{\lambda}$. Since most edges (*i.e.,* object-tag pairs) in the original bipartite graph $\mathbb{G}$ could be invalid[1], we try to prune such edges first. Then, for any edge e_k in $\mathbb{G}$ with coefficient a_{e_k}, we calculate the *z-score*, which measures a value's relationship to the mean of a group of values, in units of the standard deviation. Formally,

$$z_k = \frac{a_{e_k} - \bar{a}}{\sigma}. \tag{8}$$

A higher z-score indicates the value has larger deviation from $\bar{a}$, which means such edge is more likely to be invalid. We set a predefined threshold η. If $z_k > \eta$, the corresponding edge is regarded to be invalid and further removed from the graph (shown in Fig. 5(b)).

C. Acquiring Agreement Among Edges

As we mentioned before, the measured phase jumps when it approaches 0 or 2π because of the mod operation [12]. Therefore, if two traces always maintains a distance difference

[1]Saying an edge is invalid means that the associated tag and motion blob are almost impossible to be matched.

of $\lambda/2$ in Eqn. 2, then their phase values would be identical, which results in the ambiguity of tags. To deal with this, we propose to incorporate the RSSI attribute of the tag's backscatter signal.

In addition to the RF phase, it is also possible to gain the received signal strength indicator (RSSI) utilizing commercial RFID readers. RSSI is a measurement of the power present in a received radio signal, which is inversely proportional to the distance between the reader and tag. Compared to the phase, the collected RSSI is more sensitive to environmental surroundings, and thus prone to contain more errors. Here in this work, we mainly exploit the RSSI attribute to qualitatively analyze the distance relationship of different tags, offering a beneficial supplement to the phase. Provided with two tags' RSSI data $\mathcal{R}_i = \{(t_1, r_i[t_1]), (t_2, r_i[t_2]), \cdots\}$ and $\mathcal{R}_j = \{(t_1, r_j[t_1]), (t_2, r_j[t_2]), \cdots\}$, we compare their values element-by-element and convert the result into a vector $\overrightarrow{x_{i,j}}$ that contains only '0's and '1's. The p^{th} element in $\overrightarrow{x_{i,j}}$ is set to '1' if $r_i[t_p] > r_j[t_p]$ and '0' otherwise. Similarly, given two objects' image trace distances $\mathcal{S}_k = \{(t_1, S_k[t_1]), (t_2, S_k[t_2]), \cdots\}$ and $\mathcal{S}_l = \{(t_1, S_l[t_1]), (t_2, S_l[t_2]), \cdots\}$, we also compare their values and get a vector $\overrightarrow{y_{k,l}}$. The p^{th} element in $\overrightarrow{y_{k,l}}$ is set to '1' if $S_i[t_p] > S_j[t_p]$ and '0' otherwise.

Eventually, we propose a metric named *agreement* to evaluate the consistency between edges (object-tag pairs) in graph $\mathbb{G}$. The agreement of any two edges $e = (u_i, v_k)$ and $f = (u_j, v_l)$ is defined as the similarity of their associated vectors $\overrightarrow{x_{i,j}}$ and $\overrightarrow{y_{k,l}}$ as below:

$$\mathsf{agreement}(e, f) = \frac{\overrightarrow{1}^{\mathrm{T}} \cdot (\overrightarrow{x_{i,j}} \oplus \overrightarrow{y_{k,l}})}{|\overrightarrow{x_{i,j}}|}. \tag{9}$$

Here, $()^{\mathrm{T}}$ denotes the matrix transpose, $\oplus$ represents the exclusive OR, $\cdot$ represents the dot product operation, and the lengths of $\overrightarrow{x_{i,j}}$ and $\overrightarrow{y_{k,l}}$ are the same. The value of an *agreement* would fall inside $[0, 1]$, and a higher *agreement* indicates that the related edges are more consistent with each other when evaluated in terms of the RSSI.

D. Identifying All the Targets

After filtering invalid edges with the technique in Section V-B, suppose that there are N connected pairs in total remaining in the graph. As shown in Fig. 5(c), we can construct an $N \times N$ dimensional matrix $\mathbf{Q}$ with the computed pair-wise agreement value via Eqn. 9. In order to identify the correct mapping in the graph, our algorithm tries to select a group of edges with as high weights and agreement values as possible.

Let $\mathbf{X}$ ($N \times 1$ dimension) be the selection result where an element equals 1 if the corresponding edge is selected, and 0 if not. Here, there are two constraint conditions: 1) a motion blob can only be mapped to one RF signal, which means that if there is more than one edge that connects the same vertex in $\mathbb{U}$, they can not coexist; and 2) multiple traces can link to the same tag, because a moving target may go outside the camera's surveillance region and later come back, and in this condition, there would be more than one traces for the same tag with no overlapped time interval existing. Considering the above two constraint conditions, we further construct a compatibility matrix $\mathbf{P}$ (consists of 0 and 1), where each row represents an incompatible case. In particular, the '1's in a row indicates the related edges are mutually incompatible. Eventually, the selection of object-tag pairs can be formalized into the following optimization problem [13]:

$$\begin{aligned} &\max_{\mathbf{X}} \mu \mathbf{X}^{\mathrm{T}} \mathbf{Q} \mathbf{X} + \mathbf{W} \mathbf{X}, \\ &\mathbf{P}\mathbf{X} \leq \mathbf{1}, \mathbf{X}_i \in \{0, 1\}, \end{aligned} \tag{10}$$

where $\mathbf{W}$ is the matrix of edges' weights, and μ is a user-defined constant parameter that controls the effect of *agreement* on the result. μ is chosen to be 0.5 by default. We can adjust the value of μ according to the precision of RSSI reported by the reader in practice. As shown in Fig. 5(d), after solving the optimization through linear programming, we can finally get a mapping between motion blobs and RFID tags, and further, accomplish the goal of object identification and tracking.

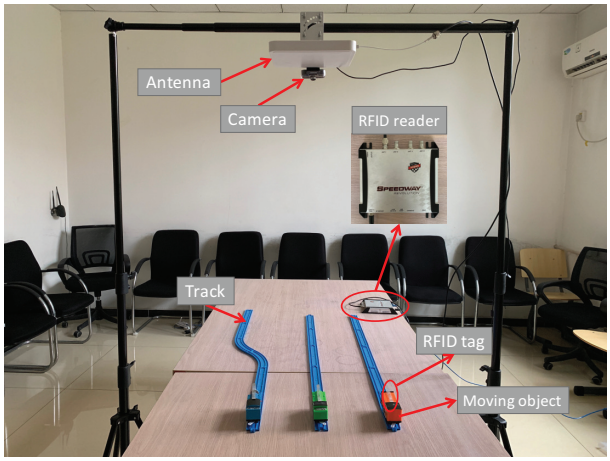


Fig. 6: Experiment setup. The RFID antenna and camera are deployed together. Tags are attached on moving objects.

E. Putting Things Together

In summary, the whole workflow of our object identification schema is outlined in Fig. 5. First, we establish a complete bipartite graph $\mathbb{G}$ where the two sides represent the moving objects and RFID tags respectively. The weight of each edge is measured by the R-squared of their associated data. A maximum-weight matching $\mathbb{M}$ is found in the graph with the Hungarian algorithm. Second, we compute a z-score for each edge in $\mathbb{G}$ utilizing the data from $\mathbb{M}$ and prune those invalid edges with z-scores above a threshold. Third, we calculate an agreement value for any two edges remaining in the graph by incorporating tags' RSSI information. Thus an inter-edge agreement matrix is acquired. Finally, we abstract a linear optimization problem with graph weights, agreement matrix, and constraint conditions. The solution to this problem is outputted as the final mapping between targets and tags. Furthermore, by combining the identification and tracking

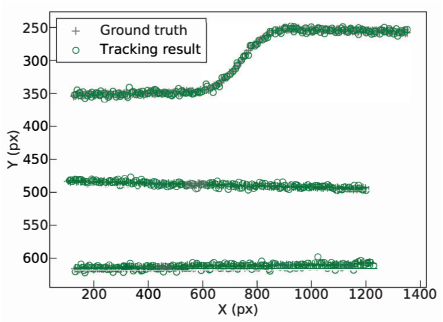
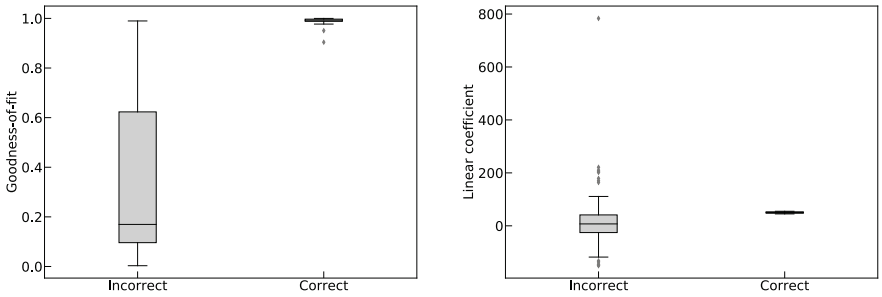


Fig. 7: Tracking result



(a) Goodness-of-fit (b) Linear coefficient

Fig. 8: Linear fitting results

results, we acquire the accurate identity and trajectory of each target in the surveillance region.

VI. Implementation & Evaluation

We implement Tagview using COTS RFID and camera equipment and conduct performance evaluation in our lab environment as shown in Fig. 6.

A. Building Prototype

Hardware. We adopt an Impinj Speedway R420 reader [14] which is compatible with EPC Gen2 standard and operates during the frequency band of $920.5 \sim 924.5$ MHz by default. The reader is connected to our host via Ethernet. One antenna with circular polarization and $8\,\mathrm{dBi}$ gain is employed, with size of $225\,\mathrm{mm} \times 225\,\mathrm{mm} \times 33\,\mathrm{mm}$. We experiment on four types of tags from Alien Corp [15], modeled "Squiggle", "Short", "Square" and "2×2". The camera we use is an AONI C30HD with 1080P resolution and frame rate of $30\,\mathrm{fps}$.

Software. Our implementation involves the LLRP (Low Level Reader Protocol) [16] to communicate with the reader. Impinj readers extend this protocol to support the ID, phase and RSSI readings of tags. The object tracking method is implemented in Python using PyTorch as the deep learning library and CUDA 9.0 as the GPU computing platform. We use a Lenovo PC to run all our algorithms and as the host to connect to the reader under LLRP. The machine equips Intel Core i5 CPU running at $2.3\,\mathrm{GHz}$ and $8\,\mathrm{GiB}$ memory.

Baseline. We emulate a mobile object via a toy train on which a tag is attached, moving on tracks of different shapes (linear or arc-shaped) with a moderate speed. The ground truth of tag-object combinations are manually collected in our experimentation.

B. Accuracy of Multi-Object Tracking

We first evaluate the multi-object tracking accuracy of Tagview in the image. For object detection purpose, we attach a predefined pattern (*e.g.,* checker pattern) onto each toy trains. Then we train the SSD model as described in Section III to detect such patterns in every image frame. To assess the final tracking performance, we lower the frame rate of the video and collect the ground truth by manually annotating the center locations of the toy trains at every video frame. Fig. 7 shows an example of tracking results collected in our experiments. There are three moving objects in the scene with different shapes of trajectories. The ground truth locations are marked as '+' in black and the tracking results output by our tracker are labeled as '$\circ$' in green. Empirical analyses indicate that the mean error distance of our tracking method is 3.76 pixels in x-axis, 3.53 pixels in y-axis and 4.82 pixels (corresponding to a $5.36\,\mathrm{mm}$ physical distance) in combined dimension with standard deviation of 3.42 pixels. We believe such mm-level accuracy is relatively high and sufficient for most applications.

C. Effectiveness of Linear Regression Model

As described in Section IV-B, we propose to use the goodness-of-fit (R-squared) of a linear model to measure the consistency between a given pair of tag and moving object. To assess its effectiveness, we carry out 50 groups of experiments. Specifically, we vary the number of tagged objects from 2 to 6, and control them to move on a desktop with a size of $1.5\,\mathrm{m} \times 3\,\mathrm{m}$. Each time we collect the phase data of attached tags and image information of moving objects. The two boxplots in Fig. 8 show the distribution of linear fitting results with regard to the R-squared and linear coefficient respectively. The 'correct' box represents the true tag-object pairs while the one labeled 'incorrect' is acquired from the false fittings. From Fig. 8(a) we observe that the correct fittings have apparently larger R-squared values compared to the false ones. This also conforms to our assumption in Section V-B that most of the edges in the maximum-weight matching are correct (*i.e.,* accord with the ground truth). Besides, from Fig. 8(b) we can see that the linear coefficients of the correct fittings exhibit a denser distribution. This is consistent with our observation V.1 that all correctly matched tag-object pairs should maintain the same linear coefficient a.

Our experiments show that if we only utilize the Hungarian algorithm (to find the maximum-weight matching), the target identification accuracy can reach 0.91 on average. But as we discussed before, it can not deal with special cases, for example, one tag relates to multiple motion traces, *etc.*

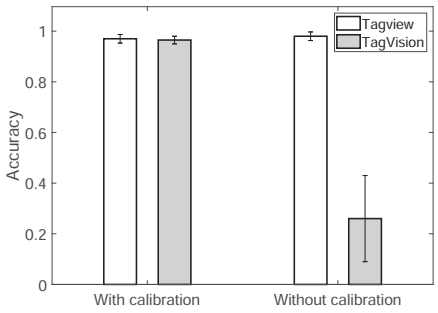


Fig. 9: Identification accuracy

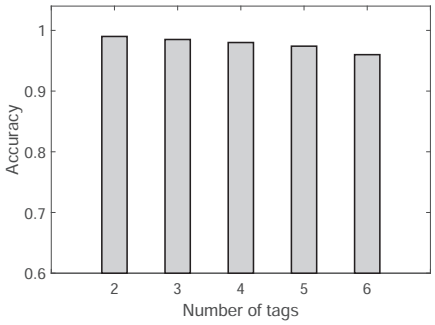


Fig. 10: Impact of tag number

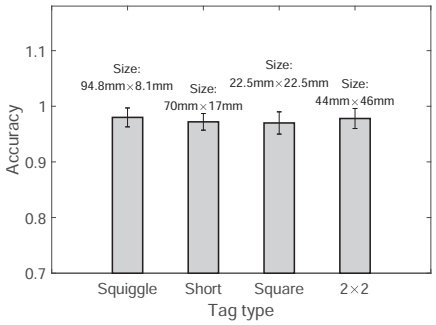


Fig. 11: Impact of diversity

D. Performance of Target Identification

1) Overall Identification Accuracy: The key to Tagview lies in our identification schema which identifies every moving target by assigning it to an optimal RF tag. To evaluate the performance of our proposed schema, we make a comparison study with prior work TagVision [3]. We consider the following two situations: a) the camera's intrinsic parameters are unknown, which means we need to calibrate the camera first; b) the intrinsic parameters are known in advance while the extrinsic parameters remain unknown. Fig. 9 plots the target *identification accuracy*, which is defined as the ratio of the number of successfully identified object traces to the total number of traces.

We find that both Tagview and TagVision achieve high precision (ratio of 0.98 on average) if the camera's parameters are known. However, if we do not perform camera calibration or the camera's relative position changes, TagVision would fail (accuracy drops significantly below 0.30) because it requires to learn the real-world trajectories of moving objects, while our system can still work in such scenario as we only need to perform image-level object tracking.

2) Impact of Tag Number: Since our system focuses on the more pervasive problem of identifying multiple targets, we further carry out experiments to evaluate Tagview's performance when there are a different number of tagged objects. We vary the number of tagged objects from 2 to 6 and plot the averaged identification accuracy in Fig. 10. It can be seen from this figure that with the number of tags increasing, the mean accuracy of identifying tags decreases a little bit, from 0.99 when there are only two tags, to about 0.96 when there are six tags. This is reasonable because, with more tagged objects, it becomes more difficult to recognize their correct traces, and thus introducing more errors. Generally speaking, our method works well at identifying tags' trajectories even when there are multiple tagged objects.

3) Impact of Tag Diversity: To study the feasibility of different types of tags, we experiment on four tag models, namely "Squiggle", "Short", "Square" and "2×2" to study the influence of tag diversity. All these tag types have different antenna sizes and shapes as depicted in Fig. 11. For each tag model, the result is averaged from 50 experiments with the number of tags varying. From the figure, we observe that the accuracies of all models maintain at a high value (more than 0.97), but there exist some differences among them. Squiggle, Short and 2×2 have very close accuracies, while Square model exhibits a slightly lower accuracy. This can be explained by the size of tag's antenna, because Square has a more compact volume (only $22.5\,\mathrm{mm} \times 22.5\,\mathrm{mm}$) 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) and thereby outputting more precise result. In our experimentation, we use model "Squiggle" by default.

4) Robustness to Separated Traces: When an object moves out of the camera's monitoring area, and then comes back after a while (note that such process may even repeat), there would be several separated traces that belong to the same ID/tag. A well-designed system could be able to deal with the above special situations. To test whether our system is able to robustly identify the separated traces with the same ID, we design the following experiment. We let several tagged objects move, and each time one of these objects may generate 2 to 4 different traces in the video. Table I presents the averaged precision and recall of target identification when the number of traces changes. From the table we observe that in all cases, the identification precision and recall maintain at a high level (≥ 0.97), which means Tagview can accurately identify each moving target, even though a tag may have multiple image traces. This also validates the robustness of our proposed identification schema.

# of separated traces	2	3	4
Precision	0.99	0.98	0.97
Recall	0.99	0.97	0.97

TABLE I: Precision and recall of identifying separated traces

	With RSSI	Without RSSI
Precision	0.97	0.57
Recall	0.97	0.55

TABLE II: Precision and recall of identifying similar phases

5) Robustness to Similar Phase Sequences: As mentioned in Section V-C, there would be ambiguity when two tags' phase sequences are similar. So Tagview incorporates the RSSI information to make a further inspection. To test our system's robustness to similar phase sequences, the following experiment is conducted. We make some objects move on a few pre-designed tracks that would generate similar phase sequences. Here the tracks should have a distance difference (to the antenna) of $\lambda/2 \approx 16\,\mathrm{cm}$ in theory. We also make a comparison study with a method that does not utilize tag's RSSI. Table II depicts the averaged precision and recall of target identification. We find that both the precision and recall drop to a big extent (the precision drops from 0.97 to 0.57 while the recall drops from 0.97 to 0.55) if we do not make use of tag's RSSI information. On the contrary, the performance of our system would not be impaired if we incorporate the RSSI. This is easy to understand because when tags have similar phase sequences, they are difficult to be distinguished purely based on the phase metric. As the RSSI is sensitive to the distance, it is pragmatic to be utilized to roughly infer distance relationship among tags.

VII. Related Work

We briefly review the literature that is related to our work in this section.

Object detection and tracking in CV. First step in the process of object tracking is to identify objects of interest in the video sequence. Pioneer methods include: a) frame differencing which works by calculating the difference between two consecutive images [17]; b) optical flow method [18] which computes the image optical flow field and does clustering according to the optical flow distribution characteristics of image; c) background subtraction [19] which is achieved by building a representation of the scene called the background model and then finding deviations from the model for each incoming frame. More recently, with the advancement of deep learning techniques, significant improvement has emerged for object detection. Overfeat [20] is one of the first modern one-stage object detectors based on fully convolutional deep networks, which performs in a multiscale sliding window fashion via a single forward pass through the CNN. Redmon *et al.* [21] propose YOLO, a unified detector casting object detection as a regression problem from image pixels to spatially separated bounding boxes and associated class probabilities. In order to preserve real-time speed without sacrificing too much detection accuracy, Liu *et al.* [4] propose SSD.

Given the object regions in the image, it is then the tracker's task to perform object correspondence from one frame to the next to generate the tracks. The main tracking categories involve: a) point tracking where objects are represented by points, and the association of the points is based on the previous object state including position and motion [22]; b) kernel tracking where kernel refers to the object shape and appearance (*e.g.,* an elliptical shape with an associated histogram) [23]; c) silhouette tracking which is performed by estimating the object region in each frame [24]. The authors in [6] propose a simple online tracking framework using Kalman filter and Hungarian method for the tracking components.

RF-based localization. An increasing amount of research in wireless domain has focused on location sensing in the past years [25]–[27]. Early attempts rely on RSSI as the fingerprint or distance ranging metric for localization purpose [28]–[30]. There is also growing interest in utilizing phase information to locate tags. One typical solution is to estimate the angle-of-arrival (AoA) which works by measuring the phase difference between the received signals at different antennas [31]–[33]. RF-IDraw [34] uses a few antenna pairs with different separations to trace the detailed shape of an RF sources trajectory. The idea of synthetic aperture radar (SAR) is also introduced to wireless localization domain. The authors in [2] realize real-time tracking of mobile tag to a very high precision by exploiting the tags mobility to build a virtual antenna array. Ubicarse [35] performs a new formulation of SAR on handheld devices twisted by their users to enable fine-grained indoor localization.

Some literature in tracking field has also explored the possibility by combining RF and CV techniques. Nick *et al.* [36] develop a camera-assisted localization algorithm based on a constrained unscented Kalman filter with tag's RSSI measurements. However, we know the RSSI is a relatively sensitive metric and thus easy to be influenced by surrounding environment. [3] proposes a fine-grained target tracking schema by assigning the identified tag's phase sequence to the most possible track of moving objects acquired with image processing. Although high accuracy has been demonstrated, it only focuses on the tracking of one tagged target, and is not convenient for repeated usage. Compared against the above methods, our work offers a more universal and user-friendly solution for the identification and tracking of multiple targets without sacrificing accuracy.

VIII. Conclusion

This work presents an RFID-based pervasive identification and tracking system for multiple objects with the aid of vision techniques. Our key innovations include leveraging a pair of properly deployed RFID antenna and video camera, and designing a novel identification schema that works with only image-level trajectory information. Experimental evaluations demonstrate that Tagview can achieve a mean target recognition accuracy of 0.98 on average with strong robustness to various circumstances. We believe our system could promote more possibilities in RFID-based tracking area.

Acknowledgment

This research is supported in part by the National Key R&D Program of China (Grants No. 2018YFB2100300 and No. 2018YFB1308601), the National Natural Science Foundation of China (Grants No. 61902212, No. 61432015 and No. 61772248), and the China Postdoctoral Science Foundation (Grants No. 2019M650683 and No. 2019M650685).

REFERENCES

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

[2] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2014.

[3] C. Duan, X. Rao, L. Yang, and Y. Liu, "Fusing RFID and computer vision for fine-grained object tracking," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2017, pp. 1–9.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 21–37.

[5] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[7] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[8] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[9] D. M. Dobkin, *The RF in RFID: UHF RFID in Practice*. Newnes, 2012.

[10] Impinj, "Speedway revolution reader application note: Low level user data support," in *Speedway Revolution Reader Application Note*, 2010.

[11] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 12, pp. 83–97, 1955.

[12] C. Duan, L. Yang, Q. Lin, Y. Liu, and L. Xie, "Robust spinning sensing with dual-RFID-tags in noisy settings," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2647–2659, 2019.

[13] D. Bienstock, "Computational study of a family of mixed-integer quadratic programming problems," *Mathematical Programming*, vol. 74, no. 2, pp. 121–140, 1996.

[14] "Impinj, Inc," http://www.impinj.com/.

[15] "Alien," http://www.alientechnology.com/tags/square.

[16] EPCglobal, "Low Level Reader Protocol (LLRP)," 2010.

[17] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.

[18] A. K. Chauhan and P. Krishan, "Moving object tracking using gaussian mixture model and optical flow," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 4, 2013.

[19] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2004.

[20] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[22] K. Shafique and M. Shah, "A noniterative greedy algorithm for multiframe point correspondence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 51–65, 2005.

[23] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 5, pp. 564–575, 2003.

[24] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1531–1536, 2004.

[25] Z. Yang, L. Jian, C. Wu, and Y. Liu, "Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 2, pp. 26:1–26:20, 2013.

[26] Z. Zhou, C. Wu, Z. Yang, and Y. Liu, "Sensorless sensing with WiFi," *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 1–6, 2015.

[27] K. Chen and G. Tan, "BikeGPS: Localizing shared bikes in street canyons with low-level GPS cooperation," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 4, pp. 1–28, 2019.

[28] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.

[29] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2012, pp. 269–280.

[30] R. Bhandari, B. Raman, K. Ramakrishnan, D. Chander, N. Aggarwal, D. Bansal, M. Choudhary, N. Moond, A. Bansal, and M. Chaudhary, "CrowdLoc: Cellular Fingerprinting for Crowds by Crowds," *ACM Transactions on Sensor Networks (TOSN)*, vol. 14, no. 1, p. 4, 2018.

[31] T. Liu, L. Yang, Q. Lin, Y. Guo, and Y. Liu, "Anchor-free backscatter positioning for RFID tags with high accuracy," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2014.

[32] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu, "STPP: Spatial-temporal phase profiling-based method for relative RFID tag localization," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 596–609, 2016.

[33] C. Duan, L. Yang, and Y. Liu, "Accurate spatial calibration of RFID antennas via spinning tags," in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2016.

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

[35] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2014, pp. 483–494.

[36] T. Nick, S. Cordes, J. Gotze, and W. John, "Camera-assisted localization of passive RFID labels," in *Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012.