TAR - Enabling Fine-Grained Targeted Advertising in Retail Stores

Xiaochen Liu
University of Southern California
liu851@usc.edu

Yurong Jiang
LinkedIn*
jiangyurong609@gmail.com

Puneet Jain
Google*
cse.puneet@gmail.com

Kyu-Han Kim
Hewlett-Packard Labs
kyu-han.kim@hpe.com

CCS CONCEPTS

• **Information systems** → **Information systems applications**; • **Networks** → **Mobile networks**; **Location based services**; • **Human-centered computing** → **Mobile computing**;

ACM Reference Format:
Xiaochen Liu, Yurong Jiang, Puneet Jain, and Kyu-Han Kim. 2018. TAR - Enabling Fine-Grained Targeted Advertising in Retail Stores. In *MobiSys '18: The 16th Annual International Conference on Mobile Systems, Applications, and Services, June 10–15, 2018, Munich, Germany.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3210240.3210342

Abstract

Mobile advertisements influence customers' in-store purchases and boost in-store sales for brick-and-mortar retailers. Targeting mobile ads has become significantly important to compete with online shopping. The key to enabling targeted mobile advertisement and service is to learn shoppers' interest during their stay in the store. Precise shopper tracking and identification are essential to gain the insights. However, existing sensor-based or vision-based solutions are neither practical nor accurate; no commercial solutions today can be readily deployed in a large store. On the other hand, we recognize that most retail stores have the installation of surveillance cameras, and most shoppers carry Bluetooth-enabled smartphones. Thus, in this paper, we propose TAR to learn shoppers' in-store interest via accurate multi-camera people tracking and identification. TAR leverages widespread camera deployment and Bluetooth proximity information to accurately track and identify shoppers in the store. TAR is composed of four novel design components: (1) a deep neural network (DNN) based visual tracking, (2) a user trajectory estimation by using shopper visual and BLE proximity trace, (3) an identity matching and assignment to recognize shopper's identity, and (4) a cross-camera calibration algorithm. TAR carefully combines these components to track and identify shoppers in real-time. TAR achieves 90% accuracy in two different real-life deployments, which is 20% better than the state-of-the-art solution.

Keywords: Shopping, Computer Vision, Mobile Sensing, Tracking, Bluetooth, Edge Computing

1 INTRODUCTION

Digital interactions influence 49% of in-store purchases, and over half of them take place on mobile devices [12]. With this growing trend, brick-and-mortar retailers have been evolving their campaigns to effectively reach people with mobile devices, showcase products, and ultimately, influence their in-store purchase. Among them, sending targeted advertisements (ads) to user's mobile devices has emerged as a frontrunner [30].

To send well-targeted information to the shopper, the retailers (and advertisers) should correctly understand customers' interest. The key to learning the customer's interest is to accurately track and recognize the customer during her stay in the store. Therefore, the retailers need a practical system for shopper tracking and identification with real-time performance and high accuracy. For example, the retailer's advertising system would require aisle-level or meter-level accuracy in tracking a shopper to infer customer's dwelling time at a certain aisle. Moreover, the advertising should be able to reflect the customer's position change fast, because people usually stay at, or walk by, a specific shelf in just a few seconds.

Some sensor-based indoor tracking metrics are invented, such as Wi-Fi localization [14, 51], Bluetooth localization [59, 84], stereo cameras [22, 26, 38, 80], and thermal sensors [28]. However, such approaches are either expensive in hardware cost or inaccurate for retail scenarios. Some commercial solutions [2, 13] send customers the entire store's information when they enter the store zone. Such promotions are coarse-grained and can hardly trigger customers' interests.

Recently, live video analytics has become a promising solution for accurate shopper tracking. Companies like Amazon Go [17] and Standard Cognition [11] use close-sourced algorithms to identify customers and track their in-store movement. The opensource community also has proposed many accurate metrics for people (customer) identification and tracking.

For people (re)identification, there are two mainstream approaches: face recognition and body feature classification. Today's face recognition solutions ([55, 66, 87]) can reach up to 95% of precision on public datasets, thanks to the advance of deep neural networks

*The work was done at Hewlett-Packard Labs.

Research was sponsored by Hewlett-Packard Labs and the Army Research Laboratory with the Cooperative Agreement Number W911NF-09-2-0053 (the ARL Network Science CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

MobiSys '18, June 10–15, 2018, Munich, Germany

ACM ISBN 978-1-4503-5720-3/18/06…$15.00
https://doi.org/10.1145/3210240.3210342

(DNN). However, the customer's face is not always available in the camera, and the face image may be blurry and dark due to poor lighting and long distance. The body-feature-based solutions [40, 47, 92] do not deliver high accuracy (< 85%) and also suffer from bad video quality.

For people tracking, the retailer needs both single-camera tracking and cross-camera tracking to understand the walking path of each customer. Recent algorithms for single-camera tracking [39, 53, 67, 88] leverage both the person's visual feature and past trajectory to track her positions in following frames. However, such algorithms cannot perform well in challenging environments, e.g., similar clothes, long occlusion, and crowded scene. Existing cross-camera tracking algorithms [46, 75, 79, 86] use the camera network's topology to estimate the cross-camera temporal-spatial similarity and match each customer's trace across cameras. Such solutions face challenges like unpredictable people movement (between the surveillance zones).

In this paper, we propose TAR to overcome the above limitations. As summarized above, existing indoor localization solutions are not accurate enough in practice and usually require the deployment of complicated and expensive equipment. Instead, this paper proposes a practical end-to-end shopper tracking and identification system. TAR is based on two fundamental ideas: *Bluetooth proximity sensing* and *video analytics*.

To infer a shopper's identity, TAR looks into Bluetooth Low Energy (BLE) signal broadcasted from the user's device. BLE has recently gained popularity with numerous emerging applications in industrial Internet-of-Thing (IoT) and home automation. Proximity estimation is one of the most common use cases of BLE beacons [4]. Apple iBeacon [18], Android EddyStone [24], and open-sourced AltBeacon [16] are available options. Several retail giants (e.g., Target, Macy's) have already deployed them in stores to create a more engaging shopping experience by identifying items in proximity to customers [27, 59, 84].

TAR takes a slightly different perspective from the above scenario in that shoppers carry BLE-equipped devices and TAR utilizes BLE signals to enhance tracking and identify shoppers. In a high level, TAR achieves identification by attaching the sensed BLE identity to a visually tracked shopper. TAR notices the pattern similarity between shopper's BLE proximity trace and her visual movement trajectories. Therefore, the identification problem converts to a trace matching problem.

In solving this matching problem, TAR encounters four challenges. First, pattern matching in real-time is challenging due to different coordinate systems and noisy trace data. TAR transforms both traces into the same coordinates with camera homography projection and BLE signal processing. Then, TAR devises a probabilistic matching algorithm that based on Dynamic Time Warping (DTW) [42] to match the patterns. To enable the real-time matching, TAR applies a moving window to match trace segments and uses the cumulative confidence score to judge the matching result.

Next, assigning the highest-ranked BLE identity to the visual trace is often incorrect. Factors like short visual traces could significantly increase the assignment uncertainty. To solve this problem, TAR uses a linear-assignment-based algorithm to correctly determine the BLE identity. Moreover, instead of focusing on a single trace, TAR looks at all visual-BLE pairs (i.e., a global view) and assigns IDs for all visual traces in a camera.

Third, a single user's visual tracking trace can frequently break upon occlusions. To solve this issue, TAR implements a rule-based scheme to differentiate ambiguous visual tracks during the assignment process and connects broken tracks, regarding each BLE ID.

Finally, it is non-trivial to track people across cameras with different camera positions and angles. Existing works [75, 79, 82, 89] either work offline or require overlapping camera coverage to handle a transition from one camera to the other. However, overlapping coverage is not guaranteed in most shops. To overcome this issue, TAR proposes an adaptive probabilistic model that tracks and identifies shoppers across cameras with little constraint.

We have deployed TAR in an office and a retail store environment, and analyzed TAR's performance with various settings. Our evaluation results show that the system achieves 90% accuracy, which is 20% better than the state-of-the-art multi-camera people tracking algorithm. Meanwhile, TAR achieves a mean speed of 11 frame-per-second (FPS) for each camera, which enables the live video analytics in practice.

The main contributions of our work are listed below:

- development of TAR, a system for multi-camera shopper tracking and identification (Sec. 3). TAR can be seamlessly integrated with existing surveillance systems, incurring minimal deployment overhead;
- introduction of four key elements to design TAR (Sec. 3);
- a novel vision and mobile device association algorithm with multi-camera support; and
- implementation, deployment, and evaluation of TAR. TAR runs in real-time and achieves over 90% accuracy (Sec. 4).

2 MOTIVATION

Retail trends: While the popularity of e-commerce continues to surge, offline in-store commerce still dominates in today's market. Studies in [23, 25] show that 91% of the purchases are made in physical shops. In addition, [15] indicates that 82% of the Millennials prefer to shop in brick and mortar stores. As online shopping evolves rapidly, it is crucial for offline shops to change the form and offer better shopping experience. Therefore, it is essential for offline retailers to understand shoppers' demands for better service given that today's customers are more informed about the items they want.

The need for shopper tracking and identification: By observing where the shopper is and how long she visits each area, retailers can identify the customer's shopping interest, and hence, provide a customized shopping experience for each people. For example, many large retail stores (e.g., Nordstrom [5], Family Dollar, Mothercare [6]) are already adopting shopper tracking solutions (e.g., Wi-Fi localization). These retailers then use the gathered data to help implement store layouts, product placements, and product promotions.

Existing solutions: Several companies [35, 72, 80] provide solutions for shopper behavior tracking by primarily using surveillance camera feeds. The solutions include features like shopper counting, the spatial-temporal distribution of customers, and shoppers' aggregated trajectory. However, they are not capable of understanding per-shopper insight (or identity). Services like Facebook [2, 13] offer targeted advertisement for retail stores. They leverage coarse-grained

location data and the shopper's online browsing history to identify the store-level information (which store is the customer visiting) and push relevant advertisements. Therefore, such solutions can hardly recognize the shopper's in-store behavior.

Camera-based tracking with face recognition can be used to infer the shoppers' indoor activities, but it also introduces several concerns – privacy, availability, and accuracy. First, the face is usually the privacy-sensitive information, and collecting such information might increase user's privacy concern (or even violation of law). Second, the face in the surveillance camera is sometimes unavailable due to various camera angles and body poses. Moreover, face recognition algorithms are known to be vulnerable to factors like image quality. Finally, face recognition requires the user's face image to train the model, which adds overhead to shoppers; asking them to submit a good face image and verifying the photo authenticity (e.g., offline ID confirmation) are not easy.

Our proposed approach: TAR adopts a vision-based tracking metric but extends it to enable shopper identification with BLE. We exploit the availability of BLE signals from the shopper's smartphone and combine the signal with vision-based technologies to achieve good tracking accuracy across cameras.

Modern smartphones equips with Bluetooth Low Energy (BLE) chip, and there are many BLE-based applications and hardware developed. A typical usage of BLE is to act as a beacon, which broadcasts BLE signal at a particular frequency. The beacon can serve as a unique identifier for the device and can be used to estimate the proximity to the receiver [69]. Our approach assumes the availability of BLE signals from shoppers, and this assumption becomes popular via incentive mechanism (e.g. mobile apps for coupons).

Therefore, in addition to our customized vision-based detection and tracking algorithms, we carefully integrate them with BLE proximity information to achieve high accuracy for tracking and identification across cameras.

In designing the system, we aim to achieve the following goals:

- **Accurate**: TAR should outperform the accuracy of existing multi-camera tracking systems. It should also be precise in distinguishing people's identity.
- **Real-time**: TAR should recognize each customer's identity in a few seconds since a shopper might be highly mobile across multiple cameras. Meanwhile, TAR should detect the appearance of people with high frame per second (FPS).
- **Practical**: TAR should not need any expensive hardware or complex deployment. It can leverage existing surveillance camera systems and the user's smartphone.

3 THE DESIGN

This section presents the design of TAR. We begin with an overview of the TAR's design and the motivating use cases of the design. Then, we explain the detailed components that address technical challenges specific to the retail environment.

3.1 Design Overview

Figure 1 depicts the design of TAR, and it consists of two major parts: 1) mobile Bluetooth Low Energy (BLE) library that enables

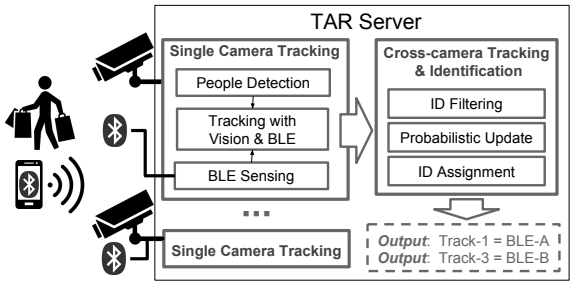


Figure 1—*System Overview for TAR*

smart devices as BLE beacons in the background, and 2) server backend that collects the real-time BLE signals and video data as well as performs customer tracking and identification. First, we assume customers usually carry their smartphones with a store application installed [34]. The store app equips with TAR's mobile library that broadcasts BLE signal as a background thread. Note that the BLE protocol is designed with minimum battery overhead [21], and the broadcasting process does not require customer's intervention. Next, TAR's server backend includes several hardware and software components. We assume each surveillance camera equips with a BLE receiver for BLE sensing. Both the camera feed and the BLE sensing data are sent to TAR for real-time processing.

TAR is composed of several key components to enable accurate tracking and identification. It has a deep neural network (DNN) based tracking algorithm (Sec. 3.3) to track users with vision trace, and then, incorporates a BLE proximity algorithm to estimate the user's movement (Sec. 3.4). In addition, TAR adopts a probabilistic matching algorithm based on Dynamic Time Warping (DTW) [42] to associate both vision and BLE data and find out the user's identity (Sec. 3.5). However, external factors such as people occlusion could harm the accuracy of sensed data and relying solely on the matching algorithm usually results in the error. To handle this issue, TAR uses a stepwise matching algorithm based on cumulative confidence score. After that, TAR devises an ID assignment algorithm to determine the correct identity from the global view (Sec. 3.5.2). As the vision-based trace might frequently break, sewing them together is essential to learning user interests. We propose a rule-based scheme to identify ambiguous user traces and properly connect them (Sec. 3.5.3). Finally, the start of the probabilistic matching process will encounter more uncertainty due to the limited trace's length. Therefore, TAR considers each user's cross-camera temporal-spatial relationship and carefully initializes its initial confidence level to improve the identification accuracy (Sec. 3.5.4).

3.2 A Use Case

Figure 2 illustrates an example of how TAR works. A grocery store is equipped with two video cameras that cover different aisles, as shown in Figure 2(a). Assume a customer with her smartphone enters the store and the app starts broadcasting BLE signal. The customer is looking for some snacks and finally finds the snack aisle. During her stay, two cameras can capture her trajectory. Briefly, camera-1 (bottom) sees the user at first and senses her BLE signals. Then the starts matching the user's visual trace to estimated BLE proximity trace. TAR maintains a confidence score for the tracked customer's BLE identity. When the user exits the camera-1 zone and enters

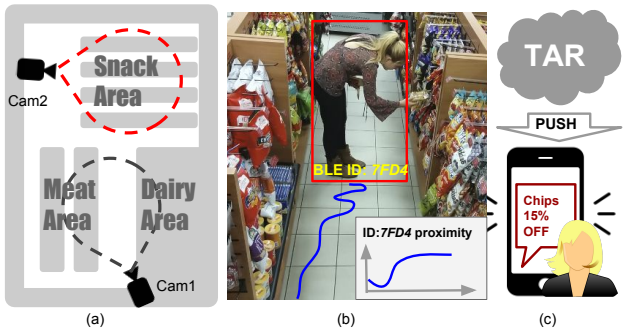


Figure 2—*A Targeted Ad Working Example in Store*

camera-2 region (top), TAR considers various factors including temporal-spatial relationship and visual feature similarity, and then, adjusts the initial confidence score for the customer in the new zone. Then, camera-2 starts its own tracking and identification progress and concludes the customer's identity (7FD4 in Figure 2(b)). TAR then continuously learns her dwell-time and fine-grained trajectory on each shelf.

In following sections, we detail core components of TAR to realize the features above and other use cases of fine-grained tracking and identification.

3.3 Vision-based Tracking (VT)

We design a novel vision-based tracking metric (VT) that consists of three components: people detection and visual feature extraction, visual object tracking, and physical trajectory estimation.

3.3.1 People Detection and Deep Visual Feature. Recent development in DNN provides us with accurate and fast people detector. It detects people in each frame and marks the detected positions with bounding boxes. Among various proposals, we choose Faster-RCNN [71] as TAR's people detector because it achieves high accuracy as well as a reasonable speed. We evaluate its performance against other options' in Sec. 4.

In addition to the detection, TAR extracts and uses the visual feature of the detected bounding box to improve inter-frame people tracking. Briefly, once a person's bounding box is detected, TAR extracts its visual feature using DNN. The ideal visual feature could accurately classify each people under different people poses and lighting conditions. Recently, DNN-based feature extractors have been proposed and outperform other features (e.g., color histogram, SIFT [64]) regarding the classification accuracy. We have evaluated the state-of-art feature extractors, including CaffeNet, ResNet, VGG16, and GoogleNet [31], and have identified that the convolution neural network (CNN) version of GoogleNet [95] delivers the best performance in the tradeoff of speed and accuracy. After that, we have further trained the model with two large-scale people reidentification datasets together (MARS [93] and DukeReID [54]), with over 1,100,000 images of 1,261 pedestrians.

3.3.2 People Tracking in Consecutive Video Frames. The tracking algorithm in TAR is inspired by DeepSort [88], a state-of-the-art online tracker. In a high level, DeepSort combines each people's visual feature with a standard Kalman filter, which matches objects based on squared Mahalanobis distance [49]. DeepSort optimizes the matching process by minimizing the cosine distance between deep features. However, it often fails when multiple people collide in a video. The cause is that the size of a detection bounding box, covering colliding people, becomes large, and the deep visual feature calculated from the bounding box cannot accurately represent the person inside.

To overcome this problem, TAR leverages the geometric relationship between objects. When multiple people are close to each other and their bounding boxes have a large intersection-over-union (IOU) ratio, TAR will not differentiate those persons using DNN-generated visual features. Instead, those people's visual traces will be regarded as "merged" until some people start leaving the group. When the bounding boxes' IOU values become lower than a certain threshold (set to 0.3), they will be regarded as "departed" and TAR will resume the visual-feature-enabled tracking.

The hybrid metric above also faces some challenges. When two users with similar color clothes come across each other, the matching algorithm sometimes fails because the users' IDs (or tracking IDs) are switched. To avoid this error, we propose a *kinematic verification* component for our matching algorithm. The idea is that people's movement is likely to be constant in a short period. Therefore, we compute the velocity and the relative orientation of each detected object in the current frame, and then compare it to existing tracked objects' velocity and orientation. This component serves as a verification module that triggers the matching only for objects whose kinematic conditions are similar. TAR avoids the confusion, as the two users above show different velocity and orientation.

The people tracking algorithm in TAR synthesizes the temporal-spatial relationship and visual feature distance to track each person (her ID) accurately. First, it adopts a Kalman filter to predict the moving direction and speed of each person (called, track), and then, predicts tracks' position in the next frame. In the next frame, TAR computes a distance between the predicted position and each detection box's position. Second, TAR calculates each bounding box's intersection area with the last few positions of each track. Larger IOU ratio means higher matching probability. Third, TAR extracts a deep visual feature (see Sec.3.3.1) of the detected object, and then, compares the feature with the track's. Here, TAR can filter out tracks with the kinematic verification, and then apply all the three matching metrics. Finally, it assigns each detection to a track. If a detection cannot match any track with enough confidence, TAR will search one frame backward to find any matched track. On the other hand, if a track is not matched for a long time (a person moves out of a camera's view), it is regarded as "missing", and hence, will be deleted.

3.3.3 Physical Trajectory Estimation. Once finishes the visual tracking, TAR then converts the results to physical trajectories by applying the homography transformation [33]. Specifically, TAR infers people's physical location by using both visual tracking results and several parameters of the camera. Assuming the surveillance cameras are stationary and well calibrated, TAR can estimate the height and the facing direction of detected objects in world coordinates. Moreover, these cameras can provide information about their current resolution and angle-of-view. With these calibration properties, TAR calculates a projective transformation matrix [33]

H that maps each pixel in the frame to the ground location in the world coordinates. As a person (or track) moves, TAR can associate its distance change with a timestamp, yielding physical trajectory.

However, the homography mapping process introduces a unique challenge; it needs to project entire pixels in a detected bounding boxes (bbox) to estimate physical distance, but the bbox size may vary frame by frame. For example, a person's bbox may cover her entire body in one frame, and then, it might only include an upper body in the next. Moreover, transforming the whole pixels in the bbox imposes an extra burden on computation. To deal with this challenge, TAR chooses a single reference pixel for each detected person, while ensuring spatial consistency of the reference pixel even in changing bbox. Specifically, TAR picks a pixel that is crossing between the bbox and ground, i.e., a person's feet position. TAR uses this bottom-center pixel of the bbox to represent its reference pixel. One may argue that the bbox's bottom may not always be a foot position (e.g., when the customers' lower body is blocked). TAR leverages the fact that a person's width and height show a ratio around 1:3. With this intuition, TAR checks whether a detected bbox is "too short" – blocked – and, if so, TAR extends the bottom side of the bbox, based on the ratio. Our evaluation shows that TAR's physical trajectory estimation achieves less than 10% of an error, even in a crowded area.

3.4 People Tracking with BLE

In addition to VT, TAR relies on BLE proximity to accurately estimate people's trajectories. We first introduce BLE beacons and then explain TAR's proximity estimation algorithm.

BLE background. BLE beacon represents a class of BLE devices. It periodically broadcasts its identifier to nearby devices. A typical BLE beacon is powered by a coin cell battery and could have $1-3$ years of lifetime. Today's smartphones support Bluetooth 4.0 protocol so they can operate as a BLE beacon (transmitter). Similarly, any device that supports Bluetooth 4.0 can be used as BLE receiver. TAR's mobile component enables a customer's smartphone as a BLE beacon. This component is designed as a library, and other applications (e.g., store app) can easily integrate it and run as a background process.

Proximity Trace Estimation. The BLE proximity trace is estimated by collecting BLE beacons' time series proximity data. Through our extensive evaluation, we select the proximity algorithm in [16] to estimate the distance from BLE beacon to the receiver. There are two ways to calculate the proximity using BLE Received Signal Strength (RSS): (1) $d = exp((E - RSS) / 10n)$ where E is transmission power (dBm) and n is the fading coefficient; (2) The beacon's transmission power ts defines the expected RSS for a receiver that is one meter away from the beacon. We denote the actual RSS as rs. Then we get $rt = \frac{rs}{ts}$. The distance can be estimated with $rt < 1.0 \; ? \; rt^{10} \; : \; c_1 rt^{c_2} + c_3$, in which c_1, c_2 and c_3 are coefficients from data regression. We implement both algorithms and compare their performance on the collected data. We find the second option is more sensitive to movement and therefore reflects the movement pattern more timely and accurately.

In practice, these coefficients depend on the receiver device manufacturers. For example, Nexus 4 and Nexus 5 use same Bluetooth chip from LG, so they have the same parameters. In TAR, we have full knowledge of our receivers, so we regress our coefficients accordingly. Since TAR also controls the beacon side, the transmission power of each beacon is known to TAR. Notice that the BLE RSS reading is inherently erroneous, so we apply the RSS noise filtering strategy similar to [44] for the original signal and then calculate the current rs with the above formula. TAR takes the time series of BLE proximity as the BLE trace for each device and its owner. We assume each customer has one device with TAR installed, while the case that one user carries multiple devices or other people's device is left for the future work.

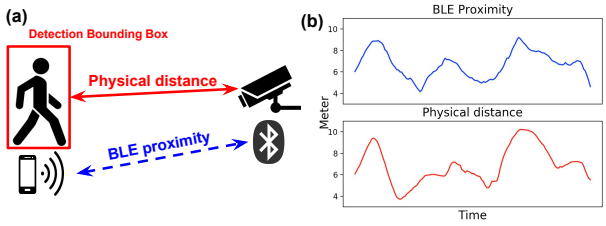
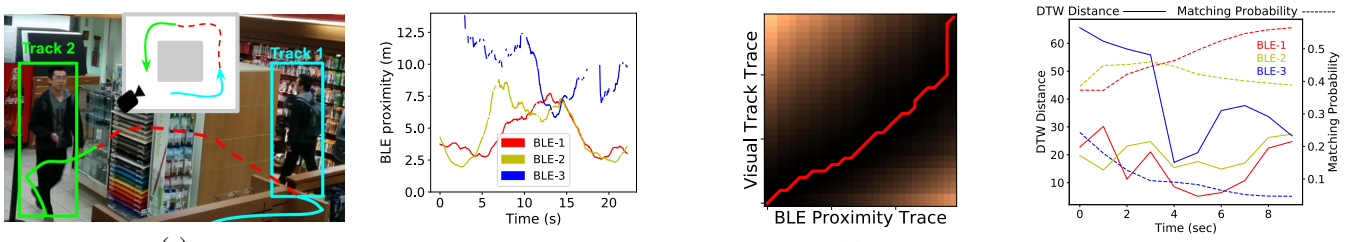


Figure 3—*Relationship between BLE proximity and physical distance*

3.5 Real-time Identity Matching

The key to learning the user's interest and pushing ads is accurate user tracking and identification. By tracking the customer, we know where she visits and what she's interested in. By identifying the user, we know who she is and whom to send the promotion. In practice, it is unnecessary to know the user's real identity. Instead, recognizing the smart devices carried by users achieves the same goal. We find the BLE universally unique ID (UUID) can serve as the identifier for the device. If we associate the BLE UUID to the visually tracked user, we will successfully identify her and learn her specific interest by looking back at her trajectories. On the other hand, we notice that for a particular user, her BLE proximity trace usually correlates with her physical movement trajectory and her visual movement. Figure 3 shows the example traces of a customer and the illustration of the BLE proximity and the physical distance. Therefore, TAR aims to associate the user's visually tracked trace to the sensed BLE proximity trace. Inspired by the observation above, We propose a similarity-based association algorithm with movement pattern matching for TAR.

3.5.1 Stepwise Trace Matching. In the matching step, we first need to decide how the traces should be matched. We notice that the BLE proximity traces are usually continuous, but the visual tracks could easily break, especially in occlusion. With this observation, we use visual tracking trace to match BLE proximity traces. The BLE trace continuity, on the other hand, can help correct the real-time visual tracking. To match the time series data, we devised our algorithm based on Dynamic Time Warping (DTW). DTW matches each sample in one pattern to another using dynamic programming. If the two patterns change together, their matched sample points will have a shorter distance, and vice versa. Therefore, shorter DTW distance means higher similarity between two traces. Based on the DTW distance, we define *confidence score* to quantify the similarity. Mathematically, assume dt_{ij} is the DTW distance between visual track v_i and BLE proximity trace b_j, the *confidence score* is $f_{ij} = exp(\frac{-dt_{ij}}{100})$.



(a) (b) (c) (d)

Figure 4—***(a)*** *Example of a visual trace;* ***(b)*** *Sensed BLE proximity traces;* ***(c)*** *DTW cost matrix for successful matching;* ***(d)*** *Matching Process Illustration.*

There are some other ways to compare the trace similarity such as Euclidean distance, cosine distance, etc. We compare the effectiveness and efficiency of these choices in Sec. 4.

DTW is a category of algorithms for aligning and measuring the similarity between two time series. There are three challenges to apply DTW to synchronize the BLE proximity and visual traces. First, DTW normally processes the two traces offline. However, both traces are extending continuously in real-time in TAR. Second, DTW relies on computing a two-dimensional warping cost matrix which has size increasing quadratically with the number of samples. Considering the BLE data's high frequency and nearly 10 FPS video processing speed, the computation overhead can increase dramatically over time. Finally, DTW calculates the path with absolute values in two sequences, but the physical movement estimated from the BLE proximity and the vision-based tracking trace are inconsistent and inherently erroneous. Computing DTW directly on their absolute values will cause adverse effects in matching.

First, to deal with the negative effect of absolute value input, TAR adopts the *data differential strategy* similar to [44, 81]. We filter out the high-frequency points in the trace and calculate the differential of current data point by subtracting the prior with time divided. Through this operation, either data sequence is independent of the absolute value and can be compared directly.

Second, a straightforward way to reduce the computation overhead is to minimize the input data size. TAR follows this path and designs a *moving window algorithm* to prepare the input for DTW. More concretely, we set a sliding window of three seconds and update the windowed data every second. We choose this window size for the balance between latency and accuracy. If the window is too short, the BLE trace and visual trace will be too short to be correctly matched. If the window is too long, we may miss some short tracks. As the window moves, TAR performs the matching process in real-time, thus solves the DTW offline issue. The window triggers the computation once the current time window is updated. Although we get the confidence score with window basis, connecting the matching windows for a specific visual track remains an issue. For example, a visual track v_i has a higher confidence to match BLE ID-1 at window 1, but BLE ID-2 at window 2. To deal with this, TAR uses *cumulative confidence* score to connect the windows for the visual track. TAR accumulates the confidence scores for consecutive windows of a visual trace and uses the cumulated the confidence score as the current confidence score for ID assignment.

ID	1	2	...	n
$Track-a$	p_{a1}	p_{a2}	...	p_{an}
$Track-b$	p_{b1}	p_{b2}	...	p_{bn}
$Track-c$	p_{c1}	p_{c2}	...	p_{cn}

Table 1—*ID-matching matrix*

We use Figure 4 as one example to demonstrate the algorithm. In this case, a customer's moving trace is shown on the top of Figure 4(a). Due to the aisle occlusion and pose change, our vision tracking algorithm obtains two visual tracks for him. Figure 4(b) shows the sensed BLE proximity during this period. TAR tries to match the visual tracked trace to one of those BLE proximity traces. Figure 4(c) shows the calculation process of DTW for a visual track and a BLE proximity trace, where the path goes almost diagonal. To illustrate our confidence score calculation process, we show the computation process for this example in Figure 4(d). The x-axis shows the time, left y-axis shows the DTW score (solid lines) for each moving window, while the right y-axis shows the cumulative confidence score (dotted lines). We can see that BLE trace 2 has better confidence score at the beginning, but falls behind the correct BLE trace 1 after four seconds.

3.5.2 Identity Assignment. To identify the user, TAR needs to match the BLE proximity trace to the correct visual track. Ideally, for each trace, the best cumulative confidence score decides the correct matching. However, there are two problems. First, as stated earlier, BLE proximity estimation is not accurate enough to differentiate some users. In practice, we sometimes see two BLE proximity traces are too similar to assign them to one user confidently. Second, visual tracks break easily in challenging scenarios, which often results in short tracks. For example, the visual track of the user in Figure 4(a) breaks in the middle, leading to two separate track traces. Although the deep feature similarity can help in some scenarios, it fails when the view angle or body pose changes. As TAR intends to learn the user interest, there needs a way to connect these intermittent visual track traces.

ID Assignment. To tackle the first challenge, TAR proposes a global ID assignment algorithm based on linear assignment [61]. TAR computes the confidence score for every track-BLE pair. At any time for one camera, all the visible tracks and their candidate BLE IDs will form a matrix called ID-matching matrix, where row i stands for track i and column j is for BLE ID j. The element (i, j) of the matrix is $Prob(BLE_{ij})$. Table 1 shows the matrix structure. Note that each

candidate ID only belongs to some of the tracks, so its matching probability is zero with other tracks.

When the matrix is ready, TAR will assign one BLE ID for the track in each row. The goal of the assignment is to maximize the total sum of confidence score. We use Hungarian algorithm [63] to solve the assignment problem in polynomial time. The assigned ID will be treated as the track's identity in the current time slot. As visual tracks and BLE proximity traces change with the time window (Sec. 3.5.1), TAR will update the assignment with updated matrix accordingly. If a track is not updated in the current window, it will be temporarily removed from the matrix as well as its candidate. When a track stops updating for a long time (> 20 sec), the system will treat the track as "terminated" and archives the last BLE ID assigned to the track.

3.5.3 Visual Track Sewing. The identity matching process is still insufficient for identity tracking in practice: the vision-based tracking technique is so vulnerable that one person's vision track may break into multiple segments. For example, upon a long period of occlusion, one person's trajectory in the camera may be split into several short tracks (see Figure 4(a)). Another case is that the customer may appear for a very short time in the camera (enters the view and quickly leaves). These short traces make the ID assignment result ambiguous as the physical distance pattern can be similar to many BLE proximity traces in that short period.

TAR proposes a two-way strategy to handle this. First, TAR tries to recognize the "ambiguous" visual track in real-time. In our design, a track will be considered as "ambiguous" when it meets either of the two rules: 1) its duration has not reached three seconds; 2) its confidence score distinction among candidate BLE IDs is vague. Explicitly, the two candidates are considered similar when the rank 2's score is more than $\geq 80\%$ of the rank 1's.

When there is an ambiguous track in assignment, TAR will first consider if the track belongs to an inactive track due to the occlusion. To verify this, TAR will search the inactive local tracks (not matched in the current window but is active within 20 seconds) and check if their assigned IDs are also top-ranked candidate IDs of the ambiguous track. If TAR cannot find such inactive tracks, that means the current track has no connection with previous tracks so the current one will be treated as a regular track to be identified with ID assignment process.

When a qualified inactive track is found, TAR will check if the two tracks have spatial conflict. The spatial conflict means the two temporally-neighbored segments locate far from each other. For example, with the same assigned BLE ID, one track v_1 ends at position P_1 and the next track v_2 starts at position P_2. Suppose the gap time between two tracks is t, and the average moving speed of T_1 is v. In TAR, T_1 and T_2 will have a spacial conflict if $|P_1 - P_2| > 5v*t$. The intuition behind is that a person cannot travel too fast from one place to another.

With the conflict check finished, TAR connects the inactive track with the ambiguous track. The trace during the gap time between two tracks is automatically fulfilled with the average speed. The system assumes that the people moves from the first track's endpoint to the second track's starting point with constant velocity during the occlusion. Then the combined track will replace the ambiguous track in the assignment matrix. After linear assignment, TAR will check if the combined track receives the same ID that is previously assigned to the inactive track. If yes, this means the track combination is successful and the ambiguous track is the extension of the inactive track. Otherwise, TAR will try to combine the ambiguous track with other qualified inactive tracks until successful ID assignment. If no combination wins, the ambiguous track will be treated as a regular track for the ID assignment process.

3.5.4 Multi-camera Calibration. In the discussion above, one problem with the matching process for the single camera is that the confidence score could be inaccurate when the tracks are short. This is due to limited amount of visual track data and the big size of candidate BLE IDs. For each visual track, we should try to minimize the number of its candidate BLE IDs. It is necessary because more candidates not only increase the processing time but also decrease the ID assignment accuracy. Therefore, TAR proposes *Cross-camera ID Selection (CIS)* to prepare the list of valid BLE IDs for each camera.

The task of CIS is to determine which BLE ID is currently visible in each camera. First, we observe that 15 meter is usually the maximum distance from the camera to a detectable device. Therefore TAR will ignore beacons with BLE proximity larger than 15 meters. However, the 15-meter scope can still cover more than 20 IDs in real scenarios. The reason is that the BLE receiver senses devices in all directions while the camera has fixed view angle. Therefore, some non-line-of-sight beacon IDs can pass the proximity filter. For example, two cameras are mounted on the two sides of a shelf (which is common in real shops). They will sense very similar BLE proximity to nearby customers while a customer can only show up in one of them.

To solve the problem, TAR leverages the positions of the camera and the shop's floorplan to abstract the camera connectivity into an undirected graph. In the graph, a vertex represents a camera, and an edge means customers can travel from one camera to another. Figure 5(a) shows a sample topology where there are four cameras covering all possible paths within the area. A customer ID must be sensed hop-by-hop. With this knowledge, TAR filters ID candidates with the following rules: 1) At any time, the same person's track cannot show up in different cameras if the cameras do not have the overlapping view. In this case, if an ID is already associated with a track in one camera with high confidence, it cannot be used as a candidate in other cameras (Figure 5(b)). 2) A customer's graph trajectory cannot "skip" node. For example, an unknown customer sensed by cam-2 must have shown up in cam-1 or cam-3, because cam-2 locates in the middle of the path from cam-1 to cam-3, and there's no other available path (Figure 5(c)). 3) The travel time between two neighbor cameras cannot be too short. We set the lower bound of travel time as 1 second (Figure 5(d)).

CIS runs as a separate thread on the TAR server. In every moving window, it collects all cameras' BLE proximity traces and visual tracks. CIS checks each BLE ID in the camera's candidate list and removes the ID if it violates any of the rules above. The filtered ID list will be sent back to each camera module for ID assignment.

4 EVALUATION

In this section, we describe how TAR works in the real scenario and evaluate each of its components.

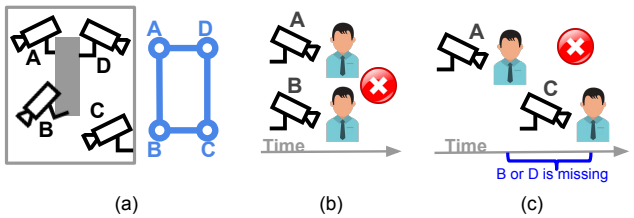


Figure 5—***(a)*** *Camera Topology;* ***(b)*** *One ID cannot show in two cameras;* ***(c)*** *BLE ID must be sensed sequentially in the network;* ***(d)*** *It takes time to travel between cameras*

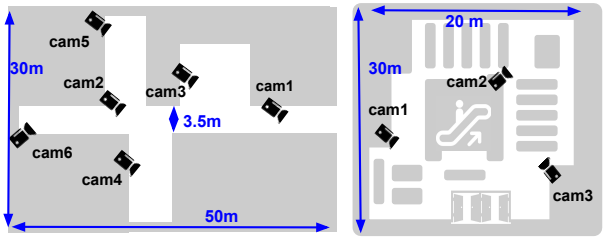


Figure 6—*Experiment Deployment Layout:* ***(a)*** *Office;* ***(b)*** *Retail store.*

Figure 7—*Same person's figures under different camera views (office).*

4.1 Methodology and Metrics

TAR Implementation. Our implementation of TAR contains three parts: the BLE broadcasting and sensing, the live video detection, and tracking and the identity matching.

BLE broadcasting is designed as a mobile library supporting both iOS and Android. TAR implements the broadcasting library with the *CoreBluetooth* framework on iOS [7] and AltBeacon [16] on Android. The BLE RSS sensing module sits in the backend. In our experiments, we use Nexus 6 and iPhone 6 for BLE signal receiving. The Bluetooth data is pushed from the devices to the server through TCP socket. In TAR, we set both the broadcasting and sensing frequency at 10 Hz.

The visual tracking module (VT) consists of a DNN-based people detector and a DNN feature extractor. One VT processes the video from one camera. TAR uses the Tensorflow version of Faster-RCNN [48, 71] as people detector and our modified GoogleNet [31] as the deep feature extractor. We train the Faster-RCNN model with VOC image dataset [52] and train the GoogleNet with two pedestrian datasets: Market-1501 [94] and DukeMTMC [96]. The detector returns a list of bounding boxes (bbox), which are fed to the feature extractor. The extractor outputs 512-dim feature vector for each bbox. We choose FastDTW [76] for DTW algorithm and its code can be downloaded from [3].

Since each VT needs to run two DNNs simultaneously, we cannot support multiple VTs on single GPU. To ensure performance, we dedicate one GPU for each VT instance in TAR, while leaving further scalability optimization to the future works. The tracking algorithm and identity matching algorithm is implemented with Python and C++. To ensure real-time process, all modules run in parallel through multi-threading.

Our server equips with Intel Xeon E5-2610 v2 CPU and Nvidia Titan Xp GPU. In the runtime, TAR occupies 5.3GB of GPU memory and processes video at around 11FPS. Double VT instances on one GPU will not overflow the memory but will reduce the FPS by around half.

As cross-camera tracking and identification require collaboration among different cameras, TAR shares the data by having one machine as the master server and running Redis cache. Then each VT machine can access the cache to upload its local BLE proximity and tracking data. The server runs cross-camera ID selection with the cached data and writes filtred ID list to each VT's Redis space.

TAR Experiment Setup. We evaluate TAR's performance by deploying the system in two different environments: an office building (Figure 6(a)) and a retail store (Figure 6(b)). We use Reolink IP camera (RLC-410S) in our setup. The test area for the office deployment is $50m \times 30m$ with the average path width of 3.5m, while the retail store is $20m \times 30m$.

We deploy six cameras in the office building as shown in the layout, and three cameras in the retail store. All the cameras are mounted at about 3m height, pointing 20°-30° down to the sidewalk. There are 20 different participants involved in the test, 12 in office deployment and 8 in retail store deployment. Besides the recruited volunteers, TAR also records other pedestrians and it captures up to 27 people simultaneously in the cameras. Each participant has TAR installed in their devices and walks around randomly based on their interest. To quantify the TAR performance, we record all the trace data in two deployment scenarios for later comparison. We've collected around 1-hour data for each deployment, including 30GB video data and 10MB BLE RSS logs. Fig.7 shows the same person's appearance in different cameras. We can see that some snapshots are dark and blurry, which makes it hard to identify people only with vision approach.

For cross-camera tracking and identification, we mainly use IDF1 Score [73], a standard metric to evaluate the performance of multi-camera tracking system. IDF1 is the ratio of correctly identified detections over the average number of ground-truth, which equals (Correctly identified people in all frames) / (All identified people in all frames). For example, if one camera records three people A, B, and C. If an algorithm returns two traces: one on A with ID=A, and another on C with ID=B. In this case, we only have one person correctly detected, so the IDF1=33%.

4.2 TAR Runtime

Before discussing our trace-based evaluation, we show the benefits of TAR's matching algorithm and optimization in the runtime.

Figure 8—*Screenshots for Cross Camera Calibration*

We first show TAR's ID assignment process [1]. In the beginning, with only detection and bbox tracking, we cannot tell the user identity. We consider the user movement estimated from the visual track and the BLE proximity traces and then apply our stepwise matching algorithm. After that, we use our ID assignment algorithm to report the user's possible identity. Although the user's identity is not correct at first, the real identity emerges as time window moves. This proves the effectiveness of our identity matching algorithm.

We also demonstrate how TAR's track sewing works in runtime[2]. As the first part of the video shows, in the case of broken visual tracks, the user may not get correctly identified after the break. By applying our track sewing algorithm, the user's tracks get correctly recognized much faster. Therefore, TAR's track sewing algorithm benefits those scenarios.

Figure 8 shows two cameras' screenshots in the office settings at different time. In this trace, one user (orange bbox) walks from camera 2 to camera 4. Meanwhile, there are around 7 BLE IDs sensed. With the user enters camera 4, TAR uses temporal-spatial relationship and deep feature distance to filter out unqualified BLE IDs, and then assigns the highest-ranked identity to the user. As shown in camera 4's screenshot, the user is correctly identified.

4.3 TAR Performance

4.3.1 Comparing with Existing Multi-cam Tracking Strategies. Figure 9(a) shows the accuracy of TAR. The y-axis represents IDF1 accuracy. As a comparison, we also evaluate the IDF1 of existing state-of-the-art algorithms from vision community:

(1) MCT+ReID: We use the work from DeepCC [75], an open-sourced algorithm that reaches top accuracy in MOT Multi-Camera Tracking Challenge [8]. The solution uses DNN-generated visual features for people re-identification (ReID) and uses single-camera tracking and cross-camera association algorithms for identity tracking. The single-camera part of DeepCC runs a multi-cut algorithm for detections in recent frames and calculates best traces to minimize the assignment cost. For cross-camera identification, it not only considers visual feature similarity but also estimates the movement trajectory of each person in the camera topology to associate two tracks, which has the similar idea of TAR in cross-camera ID selection.

[1] https://vimeo.com/246368580
[2] https://vimeo.com/246388147

(2) MCT-Only: We also tested MCMT [73], the previous work of DeepCC [75], which shares similar logic for tracking as DeepCC (both single-camera and multi-camera) but does not have DNN for people re-identification.

(3) ReID-Only: We directly run DeepCC's DNN to extract each people's visual feature in each frame and classify each person to be one of the registered users. This will show the accuracy of tracking with re-identification only.

Analysis: We can see that TAR outperforms existing best offline algorithm (MCT+ReID) by 20%. Therefore, we analyze the failures in both TAR and MCT+ReID to understand why TAR gains much higher accuracy. There are two types of failures: erroneous single-camera tracking and wrong re-identification. Note that the re-identification is BLE-vision matching in TAR's case.

As Figure 9(b) shows, the two failures have the similar contribution in TAR. In the vision-only scenario, most errors are from the re-identification process. We further break down the re-identification failures for MCT+ReID into three types: (1) multi-camera error: a person is constantly recognized as someone else in the cameras after his first appearance; (2) single-camera error: a customer is falsely identified in one camera; (3) part-of-track error: a person is wrongly recognized for part of her track in one camera. From Figure 9(b), we can see that more than half of the ReID problems are cross-camera type, which is due to the MCT module that optimizes identity assignment across cameras - if a person is assigned an ID, she will have a higher probability to get the same ID in following traces.

The root cause of the vision-based identification failure is the imperfect visual feature, which cannot accurately distinguish one person from another in some scenarios. From our observation, there are three cases that the feature extractor may easily fail: (1) blurry image; (2) partial occlusion; (3) similar appearance. Figure 9(c) demonstrates each failure case where two persons are recognized as the same customer by TAR. The figure also shows the percentage of all failure cases in the test results. We can see that the blurry and low contrast figures lead to near half of errors and the other two types account for about 40% of the failed cases.

4.3.2 Importance of Different Components in TAR. Next, we analyze each component used in TAR.

People Detection. The people detector may fail in two ways: false positive, which recognizes a non-person object as a people, and false negative, which fails to recognize a real person. For false positives, TAR could filter them out in the vision-BLE matching process. For false negatives, people occluded larger than > 80% of their bodies usually will be hardly detected by the detection model. Such false negative cases can be handled by TAR's tracking algorithm and track sewing metric, which will also be evaluated. We evaluate the performance of current state-of-the-art open-sourced people detectors using our dataset and the results are shown in Figure 11. Besides Faster-RCNN (used by TAR), we also test Mask-RCNN [56], YOLO-9000 [70], and OpenPose [43]. We can see that YOLO and OpenPose have lower accuracy although they are fast. In contrast, Mask-RCNN is very accurate but works too slow to meet TAR's requirement.

Trace matching. DTW plays the key role in matching BLE traces to vision traces. Therefore, we should understand its effectiveness

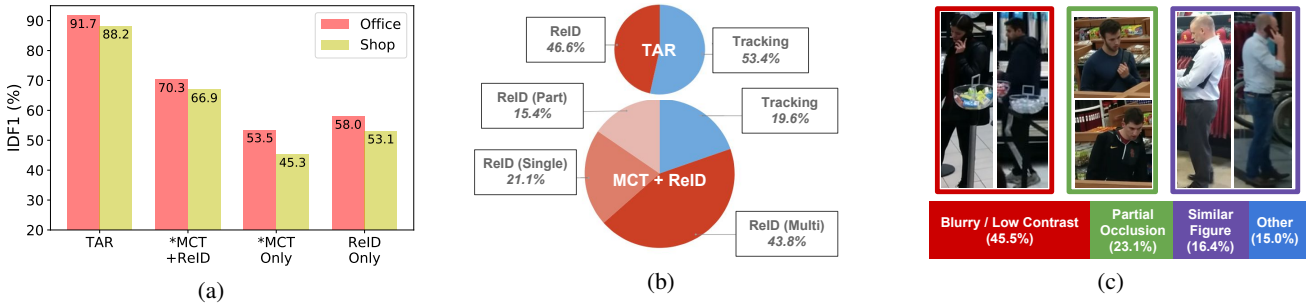


Figure 9—***(a)*** *Multi-cam tracking comparison against state-of-the-art solutions (*offline solution);* ***(b)*** *Error statistics of TAR and MCT+ReID;* ***(c)*** *Error statistics of re-identification in MCT+ReID and example images.*

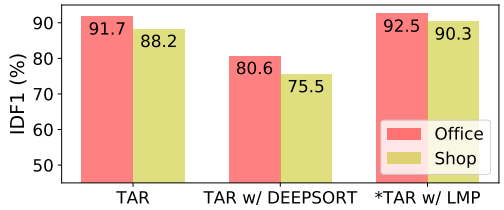


Figure 10—*Importance of Tracking Components in TAR (*offline solution).*

Similarity Metric	**Accuracy (%)**
DTW (used in TAR)	95.7
Euclidean Distance	88.0
Cosine Distance	84.9
Pearson Correlation	66.4
Spearman Correlation	72.5

Table 2—*Accuracy (ratio of correct matches) of different trace similarity metrics*

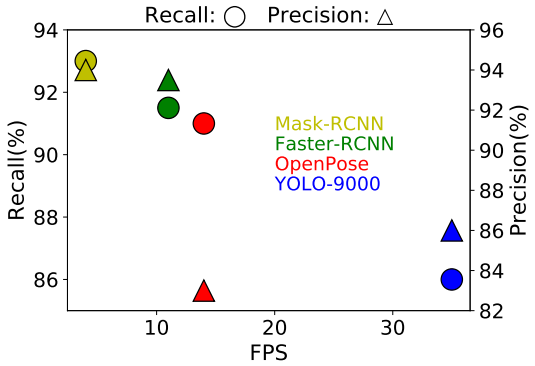


Figure 11—*Recall, precision, and FPS of state-of-the-art people detectors.*

in TAR's scenario. In the experiment, we compute the similarity between one person's walking trace and all nearby BLE traces to find the one with the highest similarity. The association process succeeds if the ground truth trace is matched, otherwise, it fails. We calculate the number of correct matchings across the whole dataset and compute the successful linking ratio. Besides DTW, we also test other metrics including Euclidean distance, cosine distance, Pearson correlation coefficient [9], and Spearman's rank correlation [10]. The average matching ratio of each method is shown in Table 2, in which DTW gets the best accuracy.

Visual Tracking. Visual tracking is crucial for estimating visual traces. As TAR develops its visual-tracking algorithm based on DeepSORT [88], we want to see TAR's performance improvement compared with existing state-of-the-art tracking algorithms. Towards this end, we replace our visual tracking algorithm with DeepSORT and LMP [85], which achieves best tracking accuracy in MOT16 challenge. LMP uses DNN for people re-identification like DeepSORT and it works offline so it can leverage posteriori knowledge of people's movement and use lifted multi-cut algorithm to assign traces globally.

We calculate the IDF1 percentage of each choice in Figure 10. We can see from the first group of bars that TAR's visual tracking algorithm clearly outperforms DeepSORT by 10%. This is because TAR's visual tracking algorithm considers several optimizations like kinematic verification, thus reduces ID switches. Moreover, TAR performs similarly with that with LMP as the tracker, which shows that our online tracking metric is comparable to the current state-of-the-art offline solution. LMP is not feasible for TAR since it works offline and slowly (0.5FPS) while our usage scenario needs real-time processing.

We compare the following modules' performance by taking away each of them from TAR and show the system accuracy change in Figure 12.

ID Assignment. An alternative solution for our ID assignment algorithm is to always choose the best (top-1) confident candidate for each track. Thus, we compare our ID assignment to the top-1 scheme and show the result in the second group of Figure 12. We can see that the top-1 scheme is almost 20% worse than TAR. The reason is that the top-1 assignment usually has the conflict error, where different visual tracks get assigned to the same ID. TAR, on the other hand, ensures the one-on-one matching, which reduces such conflicts.

Track Sewing. If we remove the track sewing optimization, a person's fragmented tracks will need much longer time to be recognized, and some of them may be matched to wrong BLE IDs. Figure 12's third group proves this point. Removing track sewing drops the accuracy for nearly 25% in the retail store dataset, which has frequent

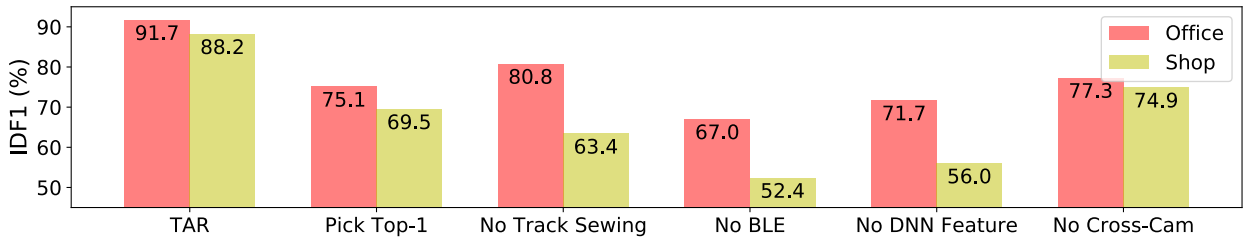


Figure 12—*Importance of Identification Components in TAR*

occlusion. In the evaluation, we find the average number of distinct tracks of the same person is 1.8, and the maximum number is 5.

BLE Proximity. Incorporating BLE proximity is the fundamental part to help track and identify users. To quantify the effectiveness of BLE proximity, we calculate the accuracy with BLE matching components removed and TAR only relies on the cross-camera association and deep visual features to identify and track each user. Figure 12's fourth group shows that the accuracy drops by 35% at most without BLE's help.

Deep Feature. The deep feature is one of the core improvements in the visual tracking algorithm. Figure 12's fifth group shows that the accuracy drops nearly 30% because removing the deep feature will cause high-frequency ID switches in tracking. In this case, it is hard to compensate the errors even with our other optimizations.

Cross Camera Calibration. Our cross-camera calibration metric contains temporal-spatial relationship and deep feature similarity across cameras. To understand the impact of this optimization, we remove the component and evaluate TAR with the same dataset. Figure 12's most right group shows a 10% accuracy drop. Without cross camera calibration, we find that the matching algorithm struggles to differentiate BLE proximity traces. In some cases, these traces demonstrate similar pattern when they move around. For example, in the retail scenario, TAR tries to recognize one user seen in camera-1 and she's leaving the store. Meanwhile, another user is also moving out but with a different direction seen in camera-2. In this case, their BLE proximity traces are hard to distinguish only with camera-1's information.

4.3.3 Robustness. Robustness is essential for any surveillance or tracking system because some part of the system might fail, e.g., one or more cameras or BLE receivers stop working. This could happen in many situations due to battery outage, camera damaged, or the lighting condition is bad. Therefore, how will those failures affect the overall performance? We focus on the system accuracy under node failures. Figure 13 shows the performance change of TAR when failure happens. Note that either the BLE failure or the video failure will cause the node failure because TAR needs both information for customer tracking. Therefore, we remove the affected nodes randomly from TAR's network to simulate the runtime failure. Figure 13 shows node failures and performance downgrades with the portion of failed nodes. We can see that TAR can still keep more than 80% accuracy with half of the nodes down. The system is robust because each healthy node could identify and track the customer by itself. The only loss from the failed node is the cross-camera part, which uses the temporal-spatial relationship to filter out invalid BLE IDs.

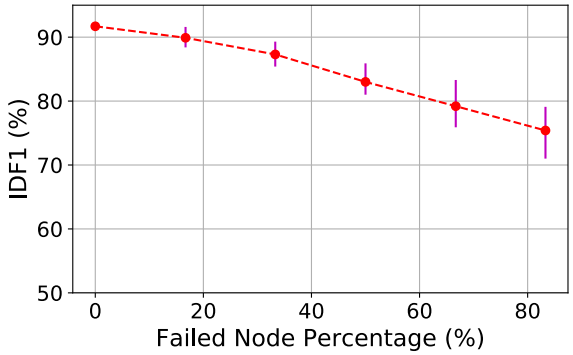


Figure 13—*Accuracy of TAR with different ratio of node failure (purple lines show the measured error).*

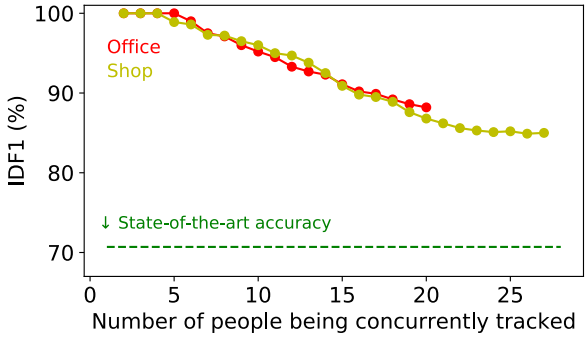


Figure 14—*Relationship between the tracking accuracy and the number of concurrently tracked people.*

We also evaluate the relationship between the number of concurrent tracked people and the tracking accuracy (shown in Figure 14). As the result shows, TAR accuracy drops as more people being tracked. The accuracy becomes stable around 85% with 20 or more people. This is because that there is no "new" trace pattern since all possible paths in each camera view are fully occupied. Therefore, adding more people will not cause more uncertainty in trace matching.

5 RELATED WORK

Mobile Coupon Deliveries Sending location-aware Ads/coupons to mobile devices has become a critical strategy for retailers [1]. Many startups [20, 29, 32, 36, 37] have been working on improving the shopping experience with mobile coupons. Among them, Urban Airship [37] is closest to TAR. It uses "point-to-radius" coordinates or indoor locations based on Wi-Fi triangulation to locate user nearby. However, Wi-Fi and other indoor localization methods require a new set of infrastructure and cannot guarantee the accuracy because they are based on proximity only. On the other hand, TAR requires little modification to existing infrastructure while providing accurate user relative position.

Indoor localization There have been plenty of works for indoor localization. Researchers have been using various devices and environmental landmarks to enhance the indoor localization accuracy [58, 62, 65, 91, 97]. Many of these designs require additional infrastructure to achieve reasonable accuracy. However, relying on indoor localization to learn user interest and send ads have two significant problems. First, most indoor localization requires a dedicated set of infrastructure. Second, indoor localization aims to help the users that actively use the device to find their location. Moreover, current indoor localization schemes are vulnerable to sophisticated indoor environments. Different from the indoor localization mechanisms, TAR does not require a separate infrastructure other than those existing surveillance cameras. It works in a passive way that users do not need use devices actively; instead, users will only get notified when TAR decides their shopping interests.

Tracking Technologies There are many tracking technologies available for people tracking. [22, 26, 38, 80] use stereo video system which utilizes camera pairs to sense 3D information of surroundings. However, the equipment is usually costly and hard to deploy. [28] uses thermal sensors to sense the existence and position of people, but its tracking accuracy can also be affected by occlusion, which makes it hard to distinguish people number. [19] uses laser and structured light to accurately infer the shape of people (usually in center-meter level), which makes them the most accurate solution for people counting. However, the short scanning range prevents the solution from continuous people tracking so other supporting solutions like cameras are needed. On the other hand, Euclid Analytics [51] and Cisco Meraki [14] have been relying on Wi-Fi MAC Address to track the customer entry and exiting the stores. However, this technology requires activation of customer Wi-Fi and suffers from location accuracy. Swirl [84] and InMarket [59] use BLE beacons to count customers, but the proximity-based approach is far from the accuracy required to track shoppers. [60] combines vision tracking with dead reckoning, which uses smartphone IMU (Inertial Measurement Unit) to estimate the user's walking speed and direction, for better user tracking accuracy. It works offline and only works on single-camera tracking. Different from the above approaches, TAR combines both vision and BLE proximity for not only tracking shoppers in large scale but also identifying shoppers.

Vision Based Tracking. Recent advances in object detection like Faster-RCNN [71], YOLO [70], and Mask-RCNN [56] have enabled accurate online detection. Therefore, tracking by detection has emerged as the leading strategy for multiple object tracking. Prior works use the global optimization that processes the entire video batches to find object trajectories. For example, three popular frameworks, network flow formulations [50, 77, 78], probabilistic graphical models [45, 57, 85] and large temporal windows (e.g. [74, 83, 90]) have been popular among them. However, due to the nature of batch processing, they are not applicable for real-time processing where no future information is available.

Recent online tracking algorithms [39, 53, 67, 88] track multiple people by matching targets in the current frame to the ones in the previous frame using their DNN-generated visual features. The algorithm works well for high-quality video as deep features are more distinguishable. For the video with low light, the visual features become hard to distinguish and the performance degrades significantly. Among the above approaches, we chose to build our tracking algorithm above DeepSORT [88] because it reaches top accuracy and works fast (> 15FPS), which is crucial for TAR scenario. Different from DeepSORT, TAR takes several optimizations mentioned in Sec.3.3 to increase the robustness against detection false negatives and occlusions.

Multi-Camera Tracking. There are some algorithms [41, 68, 82, 89] for multi-camera object tracking by knowing the positions, the directions, and internal parameters of all cameras. They also require the camera views to overlap. However, for most shops, the scene-overlapping condition is not satisfied. In contrast, TAR does not have these requirements. It utilizes various context information as well as Bluetooth signal to re-identify the objects across cameras. Some multi-camera works are designed for non-overlapping cases. Such systems [46, 75, 79, 86] leverage the visual and spatial-temporal similarity between tracks of different camera views to find the best global matching with minimum cost. However, such systems need global trajectories for best tracking accuracy, which is infeasible for online tracking. Moreover, their algorithms' accuracy entirely relies on accurate individual tracking information, i.e., people trajectories, and thereby will be affected by unreliable trackers, which are common in dense occlusion and crowded scenes.

6 CONCLUSION

We have presented TAR, a system that utilizes existing surveillance cameras and ubiquitous BLE signals to precisely identify and track shoppers and enable targeted advertising for retail stores. In TAR, we have first designed a single-camera tracking algorithm that accurately tracks people, and then extended it to the multi-camera scenario to recognize people across distributed cameras. TAR leverages BLE proximity information, cross-camera movement patterns, and single-camera tracking algorithm to achieve high accuracy of multi-camera multi-people tracking and identification. We have implemented and deployed TAR in two realistic retail-shop setting, and then conduct extensive experiments with more than 20 people. Our evaluation results demonstrated that TAR delivers high accuracy (90%) and serves as a practical solution for people tracking and identification.

Acknowledgements. We thank our shepherd Matthai Philipose and the anonymous reviewers for their valuable feedback that improved the paper's quality.

REFERENCES

[1] 3 Ways to Drive In-store Sales With Mobile. https://www.mobify.com/insights/3-ways-drive-store-sales-mobile/.

[2] Facebook Location Targeting. https://www.facebook.com/business/a/location-targeting.

[3] FastDTW. https://pypi.python.org/pypi/fastdtw.

[4] How Beacons Will Influence Billions in Us Retail Sales. http://www.businessinsider.com/beacons-impact-billions-in-reail-sales-2015-2.

[5] How Nordstrom Uses Wifi To Spy On Shoppers. https://www.forbes.com/sites/petercohan/2013/05/09/how-nordstrom-and-home-depot-use-wifi-to-spy-on-shoppers.

[6] How Retail Stores Track You Using Your Smartphone. https://lifehacker.com/how-retail-stores-track-you-using-your-smartphone-and-827512308.

[7] Ios Core Bluetooth. https://developer.apple.com/documentation/corebluetooth.

[8] MTMCT on MOT Challenge. https://motchallenge.net/data/DukeMTMCT/.

[9] Pearson Correlation Coefficient. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient.

[10] Spearman's Rank Correlation Coefficient. https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient.

[11] Standard Cognition. https://www.standardcognition.com/.

[12] What's Mobile's Influence In-Store. https://www.marketingcharts.com/industries/retail-and-e-commerce-65972.

[13] Twitter Mobile Ads. https://business.twitter.com/en/advertising/mobile-ads-companion.html, 2013.

[14] Cisco Meraki . https://meraki.cisco.com/, 2017.

[15] Accenture. https://www.accenture.com, 2017.

[16] Altbeacon. http://altbeacon.org/, 2017.

[17] Amazon Go. http://amazongo.com/, 2017.

[18] Apple IBeacon. https://developer.apple.com/ibeacon/, 2017.

[19] Bea Inc. https://www.beainc.com/en/technologies/, 2017.

[20] Best Advisor. https://www.bestadvisor.com/, 2017.

[21] Bluetooth Le: Broadcast. https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/le-broadcast , 2017.

[22] Brickstream. http://www.brickstream.com/, 2017.

[23] Deloitte. https://www2.deloitte.com, 2017.

[24] Eddystone Beacon. https://developers.google.com/beacons/, 2017.

[25] Forrester. https://go.forrester.com/, 2017.

[26] Hella. http://www.hella.com/microsite-electronics/en/Sensors-94.html, 2017.

[27] How Beacons Can Reshape Retail Marketing. https://www.thinkwithgoogle.com/articles/retail-marketing-beacon-technology.html, 2017.

[28] Irisys. http://www.irisys.net/, 2017.

[29] Moasis. http://moasis.com/, 2017.

[30] Mobile Ads. https://www.technologyreview.com/s/538731/how-ads-follow-you-from-phone-to-desktop-to-tablet/, 2017.

[31] Person Re-identification. https://github.com/D-X-Y/caffe-reid, 2017.

[32] Point Inside. https://www.pointinside.com/, 2017.

[33] Projective Transformations (homographies). http://www-prima.imag.fr/jlc/Courses/2010/ENSI3.FAI/ENSI3.FAI.S2.EN.pdf, 2017.

[34] Shopping Easier with Store App. https://corporate.target.com/article/2017/06/sean-murphy-target-app , 2017.

[35] Skyrec. http://www.skyrec.cc, 2017.

[36] Thumbvista. https://thumbvista.com/, 2017.

[37] Urban Airship. https://www.urbanairship.com/, 2017.

[38] Xovis. https://www.xovis.com/en/xovis/, 2017.

[39] Bae, S.-H., and Yoon, K.-J. Confidence-based Data Association and Discriminative Deep Appearance Learning for Robust Online Multi-object Tracking. *IEEE transactions on pattern analysis and machine intelligence 40*, 3 (2018), 595–610.

[40] Bai, S., Bai, X., and Tian, Q. Scalable Person Re-identification on Supervised Smoothed Manifold. In *CVPR* (2017).

[41] Berclaz, J., Fleuret, F., Turetken, E., and Fua, P. Multiple Object Tracking using K-shortest Paths Optimization. *IEEE transactions on pattern analysis and machine intelligence 33*, 9 (2011), 1806–1819.

[42] Berndt, D. J., and Clifford, J. Using Dynamic Time Warping to Find Patterns in Time Series. In *KDD workshop* (1994), vol. 10, Seattle, WA, pp. 359–370.

[43] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime Multi-person 2d Pose Estimation using Part Affinity Fields. In *CVPR* (2017).

[44] Chen, D., Shin, K. G., Jiang, Y., and Kim, K.-H. Locating and Tracking Ble Beacons with Smartphones.

[45] Chen, J., Sheng, H., Zhang, Y., and Xiong, Z. Enhancing Detection Model for Multiple Hypothesis Tracking. In *Conf. on Computer Vision and Pattern Recognition Workshops* (2017), pp. 2143–2152.

[46] Chen, W., Cao, L., Chen, X., and Huang, K. An Equalized Global Graph Model-based Approach for Multicamera Object Tracking. *IEEE Transactions on Circuits and Systems for Video Technology 27*, 11 (2017), 2367–2381.

[47] Chen, W., Chen, X., Zhang, J., and Huang, K. Beyond Triplet Loss: a Deep Quadruplet Network for Person Re-identification. In *Proc. CVPR* (2017).

[48] Chen, X., and Gupta, A. An Implementation of Faster Rcnn with Study for Region Sampling. *arXiv preprint arXiv:1702.02138* (2017).

[49] De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. L. The Mahalanobis Distance. *Chemometrics and intelligent laboratory systems 50*, 1 (2000), 1–18.

[50] Dehghan, A., and Shah, M. Binary Quadratic Programing for Online Tracking of Hundreds of People in Extremely Crowded Scenes. *IEEE transactions on pattern analysis and machine intelligence 40*, 3 (2018), 568–581.

[51] Euclid Analytics . http://euclidanalytics.com/, 2017.

[52] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[53] Fagot-Bouquet, L., Audigier, R., Dhome, Y., and Lerasle, F. Improving Multi-frame Data Association with Sparse Representations for Robust Near-online Multi-object Tracking. In *European Conference on Computer Vision* (2016), Springer, pp. 774–790.

[54] Gou, M., Karanam, S., Liu, W., Camps, O., and Radke, R. J. Dukemtmc4reid: A Large-scale Multi-camera Person Re-identification Dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (July 2017).

[55] Hayat, M., Khan, S. H., Werghi, N., and Goecke, R. Joint Registration and Representation Learning for Unconstrained Face Identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[56] He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask R-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on* (2017), IEEE, pp. 2980–2988.

[57] Henschel, R., Leal-Taixé, L., Cremers, D., and Rosenhahn, B. Improvements to Frank-wolfe Optimization for Multi-detector Multi-object Tracking. *CoRR* (2017).

[58] Iliev, N., and Paprotny, I. Review and Comparison of Spatial Localization Methods for Low-power Wireless Sensor Networks. *IEEE Sensors Journal 15*, 10 (2015), 5971–5987.

[59] Inmarket. https://inmarket.com/, 2017.

[60] Jiang, W., and Yin, Z. Combining Passive Visual Cameras and Active Imu Sensors to Track Cooperative People. In *Information Fusion (Fusion), 2015 18th International Conference on Information Fusion* (2015), IEEE, pp. 1338–1345.

[61] Jonker, R., and Volgenant, A. A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems. *Computing 38*, 4 (1987), 325–340.

[62] Kempke, B., Pannuto, P., Campbell, B., and Dutta, P. Surepoint: Exploiting Ultra Wideband Flooding and Diversity to Provide Robust, Scalable, High-fidelity Indoor Localization. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM* (2016), ACM, pp. 137–149.

[63] Kuhn, H. W. The Hungarian Method for the Assignment Problem. *Naval research logistics quarterly 2*, 1-2 (1955), 83–97.

[64] Lowe, D. G. Distinctive Image Features From Scale-invariant Keypoints. *International journal of computer vision 60*, 2 (2004), 91–110.

[65] Ma, Y., Hui, X., and Kan, E. C. 3d Real-time Indoor Localization Via Broadband Nonlinear Backscatter in Passive Devices with Centimeter Precision. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (2016), ACM, pp. 216–229.

[66] Masi, I., Rawls, S., Medioni, G., and Natarajan, P. Pose-aware Face Recognition in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016).

[67] Milan, A., Rezatofighi, S. H., Dick, A. R., Reid, I. D., and Schindler, K. Online Multi-target Tracking Using Recurrent Neural Networks. In *AAAI* (2017), pp. 4225–4232.

[68] Nithin, K., and Brémond, F. Globality–locality-based Consistent Discriminant Feature Ensemble for Multicamera Tracking. *IEEE Transactions on Circuits and Systems for Video Technology 27*, 3 (2017), 431–440.

[69] BLE Proximity Technologies. http://community.silabs.com/t5/Official-Blog-of-Silicon-Labs/How-to-Determine-Bluetooth-BLE-Beacon-Proximity/ba-p/173638, 2017.

[70] Redmon, J., and Farhadi, A. Yolo9000: Better, Faster, Stronger. *arXiv preprint* (2017).

[71] Ren, S., He, K., Girshick, R., and Sun, J. Faster R-cnn: Towards Real-time Object Detection with Region Proposal Networks. In *Advances in neural information processing systems* (2015), pp. 91–99.

[72] Retail Next. https://retailnext.net/en/home/, 2017.

[73] Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C. Performance Measures and a Data Set for Multi-target, Multi-camera Tracking. In *European Conference on Computer Vision* (2016), Springer, pp. 17–35.

[74] Ristani, E., and Tomasi, C. Tracking Multiple People Online and in Real Time. In *Asian Conference on Computer Vision* (2014), Springer, pp. 444–459.

[75] Ristani, E., and Tomasi, C. Features for Multi-target Multi-camera Tracking and Re-identification. *arXiv preprint arXiv:1803.10859* (2018).

[76] Salvador, S., and Chan, P. Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intelligent Data Analysis 11*, 5 (2007), 561–580.

[77] Schulter, S., Vernaza, P., Choi, W., and Chandraker, M. Deep Network Flow for Multi-object Tracking. *arXiv preprint arXiv:1706.08482* (2017).

[78] Shitrit, H. B., Berclaz, J., Fleuret, F., and Fua, P. Multi-commodity Network Flow for Tracking Multiple People. *IEEE transactions on pattern analysis and machine intelligence 36*, 8 (2014), 1614–1627.

[79] Shiva Kumar, K., Ramakrishnan, K., and Rathna, G. Inter-camera Person Tracking in Non-overlapping Networks: Re-identification Protocol and On-line Update. In *Proceedings of the 11th International Conference on Distributed Smart Cameras* (2017), ACM, pp. 55–62.

[80] Shoppertrak. https://www.shoppertrak.com, 2017.

[81] Shu, Y., Shin, K. G., He, T., and Chen, J. Last-mile Navigation using Smartphones. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (2015), ACM, pp. 512–524.

[82] Solera, F., Calderara, S., Ristani, E., Tomasi, C., and Cucchiara, R. Tracking Social Groups Within and Across Cameras. *IEEE Transactions on Circuits and Systems for Video Technology* (2016).

[83] Solera, F., Calderara, S., Ristani, E., Tomasi, C., and Cucchiara, R. Tracking Social Groups Within and Across Cameras. *IEEE Transactions on Circuits and Systems for Video Technology 27*, 3 (2017), 441–453.

[84] Swirl. http://www.swirl.com/, 2017.

[85] Tang, S., Andriluka, M., Andres, B., and Schiele, B. Multiple People Tracking by Lifted Multicut and Person Re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 3539–3548.

[86] Tesfaye, Y. T., Zemene, E., Prati, A., Pelillo, M., and Shah, M. Multi-target Tracking in Multiple Non-overlapping Cameras using Constrained Dominant Sets. *arXiv preprint arXiv:1706.06196* (2017).

[87] Tran, L., Yin, X., and Liu, X. Disentangled Representation Learning Gan for Pose-invariant Face Recognition. In *CVPR* (2017), no. 6.

[88] Wojke, N., Bewley, A., and Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. *arXiv preprint arXiv:1703.07402* (2017).

[89] Xu, Y., Liu, X., Qin, L., and Zhu, S.-C. Cross-view People Tracking by Scene-centered Spatio-temporal Parsing. In *AAAI* (2017), pp. 4299–4305.

[90] Yang, E., Gwak, J., and Jeon, M. Multi-human Tracking using Part-based Appearance Modelling and Grouping-based Tracklet Association for Visual Surveillance Applications. *Multimedia Tools and Applications 76*, 5 (2017), 6731–6754.

[91] Yang, Z., Wang, Z., Zhang, J., Huang, C., and Zhang, Q. Wearables Can Afford: Light-weight Indoor Positioning with Visible Light. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services* (2015), ACM, pp. 317–330.

[92] Zhao, H., Tian, M., Sun, S., Shao, J., Yan, J., Yi, S., Wang, X., and Tang, X. Spindle Net: Person Re-identification with Human Body Region Guided Feature Decomposition and Fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[93] ZHENG, L., BIE, Z., SUN, Y., WANG, J., SU, C., WANG, S., AND TIAN, Q. Mars: A Video Benchmark for Large-scale Person Re-identification. In *European Conference on Computer Vision* (2016), Springer, pp. 868–884.

[94] ZHENG, L., SHEN, L., TIAN, L., WANG, S., WANG, J., AND TIAN, Q. Scalable Person Re-identification: A Benchmark. In *Computer Vision, IEEE International Conference on* (2015).

[95] ZHENG, Z., ZHENG, L., AND YANG, Y. A Discriminatively Learned Cnn Embedding for Person Re-identification. *arXiv preprint arXiv:1611.05666* (2016).

[96] ZHENG, Z., ZHENG, L., AND YANG, Y. Unlabeled Samples Generated by Gan Improve the Person Re-identification Baseline in Vitro. In *Proceedings of the IEEE International Conference on Computer Vision* (2017).

[97] ZHU, S., AND ZHANG, X. Enabling High-precision Visible Light Localization in Today's Buildings. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (2017), ACM, pp. 96–108.