Towards Context Address for Camera-to-Human Communication

Siyuan Cao, Habiba Farrukh, He Wang
Department of Computer Science, Purdue University, West Lafayette, IN, USA
{cao208, hfarrukh, hw}@purdue.edu

***Abstract*—Although existing surveillance cameras can identify people, their utility is limited by the unavailability of any direct camera-to-human communication. This paper proposes a real-time end-to-end system to solve the problem of digitally associating people in a camera view with their smartphones, without knowing the phones' IP/MAC addresses. The key idea is using a person's unique "context features", extracted from videos, as its sole address. The context address consists of: motion features, e.g. walking velocity; and ambience features, e.g. magnetic trend and Wi-Fi signal strengths. Once receiving a broadcast packet from the camera, a user's phone accepts it only if its context address matches the phone's sensor data.**

We highlight three novel components in our system: (1) definition of discriminative and noise-robust ambience features; (2) effortless ambient sensing map generation; (3) a context feature selection algorithm to dynamically choose lightweight yet effective features which are encoded into a fixed-length header. Real-world and simulated experiments are conducted for different applications. Our system achieves a sending ratio of 98.5%, an acceptance precision of 93.4%, and a recall of 98.3% with ten people. We believe this is a step towards direct camera-to-human communication and will become a generic underlay to various practical applications.

***Index Terms*—surveillance camera, ID association, communication, context feature, human addressing**

I. Introduction

Video analytics has enabled widespread applications ranging from security surveillance to business intelligence [1]. Although with existing analytic algorithms, a camera can identify and track people under surveillance, its potential is not fully explored without any direct communication from the camera to people. Such communication requires an affiliation between a person and its phone, which serves as the person's unique identity for personalized message delivery from the camera. In this paper, we aim at solving the problem of digitally associating people in the camera view with their smartphones without knowing the phones' IP/MAC addresses.

The capability of sending customized messages to a specific person in a camera view can intelligently enhance public safety and daily life quality. Imagine a person on a street is being followed by someone with a suspicious behavior (shown in Figure 1(a)). Potential crimes can be prevented by informing the person about the threat. As shown in Figure 1(b), retailers like Walmart, Target, etc. can improve customers' experience by delivering targeted ads and coupons in real time, according to their interests or in-store behavior. Similarly, museums or galleries (Figure 1(c)) can provide an interactive experience to visitors by introducing interesting facts relevant to the exhibits of their interests, e.g. when a visitor points to an exhibit, through customized messages. Despite having an operator (be it a human or AI agent) monitoring the surveillance feed, the aforementioned benefits can happen only if a person can receive messages from the camera.

One may argue: Why not simply ask people to register with a face photo and then employ face recognition on the surveillance video? The main reason is that faces are not always visible due to facing direction or limited camera resolution. Moreover, many people express discomfort with uploading their profile photos, given that it may become their permanent identifier [2]. Some cities even banned the use of face recognition [3]. Another way of targeted message delivery is to add short-range communication links (e.g. acoustics, light and Bluetooth 5.1) and send messages once people approach a beacon. The deployment and maintenance of the beacons are costly. Also, message sending is simply triggered by relative distances and is not related to any contextual information (e.g. a person's behavior or surrounding events), which means these methods are not able to pinpoint an individual in a group.

Prior works have explored some schemes for human ID association involving cameras and sensors. ID-Match [4] requires wearing RFID tags and assigns a unique ID to each individual in a Kinect camera view. [5] associates people in a camera view with accelerometer readings from sensors worn on their belts. Insight [6] demonstrates that a person can be recognized using its motion patterns and clothing colors as a temporary fingerprint. However, [4], [5] require extra hardware and [4], [6] need the users to register beforehand. Also, neither of [5], [6] implements a real-time system for applications requiring direct camera-to-human communication. PHADE [7] uses walking behavior as people's temporary communication address, suffering from large packet overhead since it transmits large coefficient matrices. TAR [8] uses Bluetooth proximity sensing to associate IDs and deliver targeted advertisements in retailers. It requires Bluetooth on all users' smartphones to be on and continuously broadcast BLE signals, which raises severe privacy concerns. Moreover, [5], [8] rely on a single type of feature which is not suitable for various scenarios.

The key idea of our work is enabling camera-to-human communication using a person's *context features* as its address. The context address consists of two types of features: (1) *motion features*, e.g. walking velocity; and (2) *ambience features*, e.g. magnetic trend and Wi-Fi signal strengths in user's trajectory history. This paper pursues to utilize the diversity in these con-

(a) (b) (c)

Fig. 1. Application scenarios. (a) Send alerts to a pedestrian about potential threats. (b) Deliver ads or coupons to a customer based on his in-store shopping behavior. (c) Introduce an exhibit to a visitor when she points to it.

text features as well as the consistency within these features and mobile sensor data. We design an addressing scheme such that on the server side, based on pedestrian tracking results and ambient sensing maps (containing magnetometer and Wi-Fi data), the context features are determined for each individual in the region covered by cameras. Among these extracted features, the ones that maximize the differentiation between the target individual and the rest of the people are selected to serve as the target's context address. This context address is compressed and added as a new header in the application layer and is used to determine the destination of the packet. The server then broadcasts the packet. On the client side, upon receiving a broadcast packet, a user's phone generates corresponding features from its sensor data and compares them with the context address in the packet. If the matching score is above a threshold, the message is indeed targeted for that particular user and is relayed to the application on this phone.

When translating this idea into a functional system, we face three challenges. (1) Defining context features is nontrivial since they need to maintain both distinguishability among a group of people and some tolerance to inevitable signal noises. (2) It is hard to build and update ambient sensing maps efficiently. Simple methods, like war-driving, are not feasible due to extra and repeated human effort. (3) Selecting an optimal set of features that is discriminative and with a limited payload overhead is challenging. Ideally, these features should fit into a fixed-length header, without affecting the space available for data within a normal packet.

This paper tackles these challenges one step at a time. We present an end-to-end real-time system for camera-to-human communication based on context address, using Google Pixel XL as clients and Samsung Galaxy S5 as IP cameras. The server is designed in a pipelined and parallel manner, running on three PCs with dual NVIDIA GTX 1080 Ti SLI. We evaluate the utility and accuracy of context-based addressing from a real-world experiment in a mimic art gallery. Messages are broadcast to ten users with a sending ratio of 98.5%, an acceptance precision of 93.4%, and a recall of 98.3%. A simulated experiment in a retail store shows that the system is also feasible when scaling to a practical scenario with denser people (about 50) and more complicated environments. The context address header is always compressed to under 40 bytes, same as the length of an IPv6 header.

The main contributions are summarized below:

- Develop a novel context-based addressing scheme for camera-to-human communication, using motion and ambience features. This enables discriminating an individual from a group and sending targeted messages to it.
- Define noise-robust ambience features and an effortless way to generate ambient sensing maps with no war-driving.
- Introduce an effective context selection algorithm to dynamically choose discriminative and low-cost features.
- Implement and evaluate our system in both real-world and simulated experiments. It runs in real time and achieves high performance.

II. System Overview

Figure 2 depicts an overview of our system. Multiple cameras continuously monitor a public area and stream the video feed to a server. Upon receiving a video frame, the server conducts pedestrian detection [9]–[12] and stores the frame with its detection responses into a buffer. Once enough frames are accumulated, the detection responses in consecutive frames are associated into tracklets, each representing an individual in the camera view.

Context features are then extracted for each person from these tracklets. Each feature is either based on the person's motion pattern (e.g. whether it is walking or not at a certain timestamp) or ambience (e.g. magnetic trend in its trajectory history). The motion pattern can be directly generated from the visual tracklets, while ambience relies on maps built from magnetometer reading in gravity direction and Wi-Fi data contributed by volunteer users. Dependent on the target of the application-specific message, the context features, which can effectively distinguish the target from the rest, are selected as the context address for that person. The selected features have various lengths and formats according to their types. To reduce the payload overhead and maintain consistency in packet structure, the context features are organized and encoded into a fixed-length header. As shown in Figure 3, the context address header may also include solicitation, which requests the target user to voluntarily upload its recent magnetometer and Wi-Fi data. This data is later used to update the ambient sensing maps. The context address header is combined with a message and put into the application layer of a network packet while the destination IP/MAC address is set to all one's. The packet is then broadcast through UDP over Wi-Fi. The jobs conducted at the server are separated into stages and accomplished in a pipelined and parallel manner [7] for the system to work in

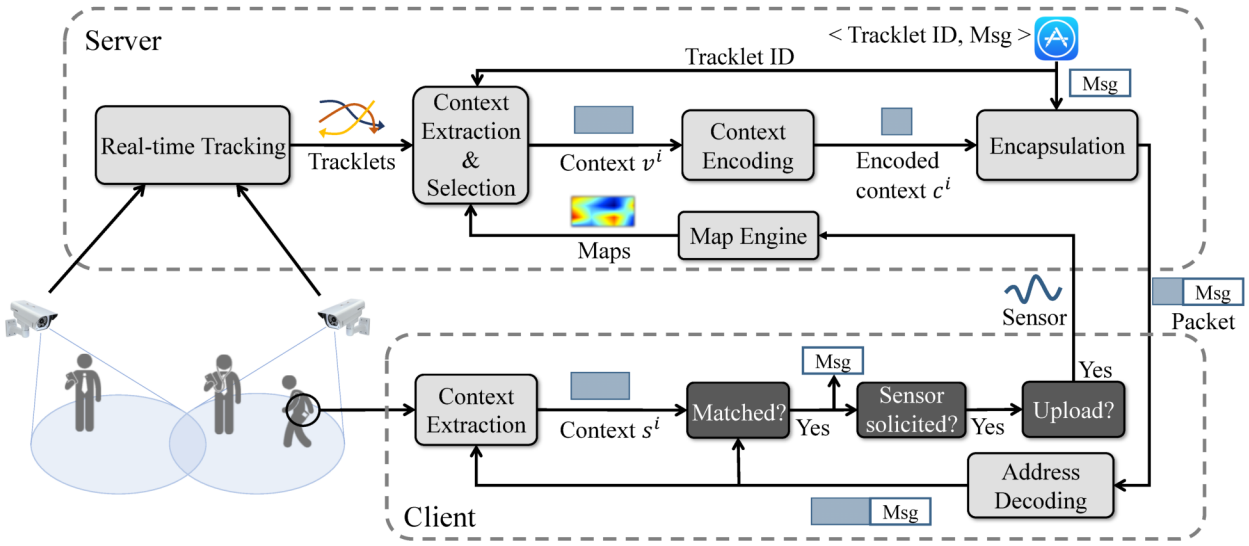


Fig. 2. System overview.

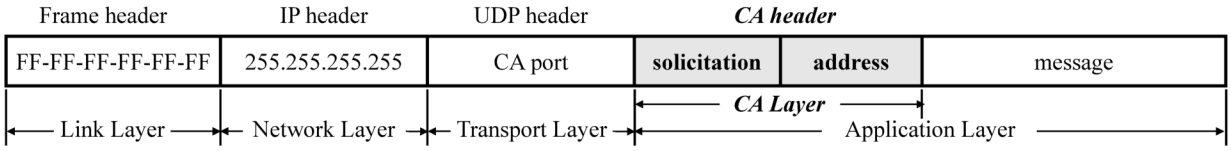


Fig. 3. Network packet with proposed context address (CA) layer.

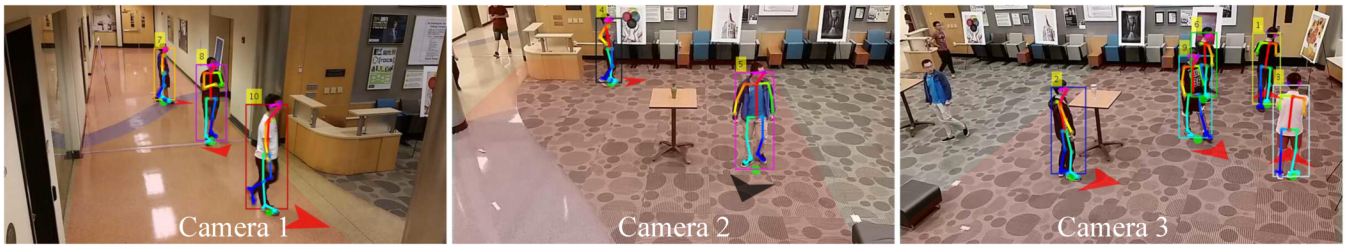


Fig. 4. Tracking result in a real-life scenario with three cameras. It is stable and accurate despite having mutual occlusions (e.g. tracklet 6).

real time. This guarantees the messages to be sent out with a short and constant delay.

On the other hand, clients passively listen to all broadcast messages and locally decide whether a message is destined for them or not. Once a smartphone carried by a person receives a broadcast packet, it extracts the corresponding motion and ambience features from its sensor readings according to the decoded context address in the packet. By comparing each feature in the context address with the smartphone's sensor readings, an overall matching score is calculated and used to decide if the message should be accepted. If accepted, the message is passed on to the upper-level applications. Also, if there is a solicitation in the header, the client may voluntarily upload the requested sensor readings to the server. This uploaded data facilitates magnetic trend and Wi-Fi map generation in the map engine on the server.

III. System Design

This section describes the design details of each component of the system beginning with the pedestrian tracking across multiple cameras, followed by the context extraction and selection process. Then it describes the structure of the context header and the matching schemes on the client side for receiving messages. Finally, the ambient sensing map generation is explained.

A. Real-time Multi-camera Human Tracking

To track people through multiple cameras in real time, we employ a pipelined and parallel scheme proposed in [7]. A state-of-the-art human pose detector, OpenPose [12], is first used for pedestrian detection. *Association Based Tracking* (ABT) [13]–[17] conducts low-level association between detection responses from neighboring frames of the streamed video. The tracklets are extended via *Category Free Tracking* (CFT) [18]–[20] and Kalman filter is applied to form local tracklets representing each person in a camera view. The local tracklets from all cameras are eventually merged into global tracklets in the entire covered area. An example of tracking results from our experiments is shown in Figure 4.

B. Context Extraction

Context features that qualify for address matching between videos and sensors should be: (1) distinguishing, i.e. having rich diversity among different people; (2) reliable, i.e. being consistent between the two sides to validate matching. We

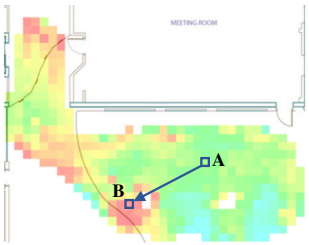


Fig. 5. Increasing Magnetic Trend between two locations.

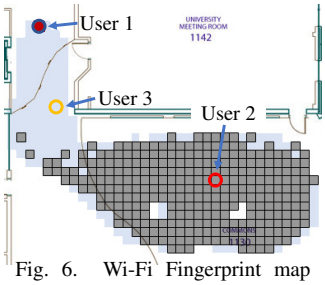


Fig. 6. Wi-Fi Fingerprint map for a reference position (shown by solid red).

define the context features, which consist of motion and ambience features. Since the context address is added into the packet payload, we encode the features to reduce the overhead.

1) Motion Features: Since the tracking process generates locations of each person, a person's velocity can be computed by applying Kalman filter on its locations [21], [22], and further extracted into motion features with no extra computation cost. We adopt the motion features defined in [7] and briefly describe them for the sake of completeness.

Moving Or Not: On the video side, from the velocity magnitude, it is straightforward to determine whether a person is *Moving Or Not*. On the sensor side, sensed acceleration is first projected onto gravity and its variance, within two seconds, is calculated. If the variance is above a predefined threshold, we can mark Moving or Not as "Yes".

Relative Rotation: We define *Relative Rotation* as the difference between a person's walking directions at the beginning and the end of a motion period. On the sensor side, rotation rates obtained from gyroscope are first projected onto the gravity and then integrated into Relative Rotation. For comparison, we define an adaptive threshold ($= Kl + B$) to compensate gyroscope drift, where l is the time length of the motion period and K, B are parameters preset to $1^\circ/s$ and 25°, respectively. If the rotation difference is within the threshold, we take it as a match.

2) Ambience Features: The visual tracking incidentally generates a user's location at each timestamp, which can be used with ambient sensing maps to extract location-related ambience features. For now, we assume that the server has a Magnetic Trend map and a Wi-Fi Fingerprint map, and will explain why we select these feature types and how we generate the maps in Section III-F.

Magnetic Trend: *Magnetic Trend* represents the difference between magnetometer readings in gravity direction at any two locations. For each pair of different locations, the difference is represented by a normal distribution using its mean (μ) and standard deviation (σ) and stored in Magnetic Trend map. Figure 5 shows an example of increasing Magnetic Trend (i.e. $\mu > 0$) from location A to B, where the blocks are different locations and red indicates larger projected magnetometer readings. When a person moves from one location to another during a motion period, a (μ, σ) pair is extracted by looking up Magnetic Trend map with its two locations and used as a candidate feature.

On the sensor side, the phone periodically samples the magnetometer and gravity readings. The 3D magnetometer readings are first projected to the current gravity to eliminate the influence of the phone pose. The projected magnetometer readings are then used to calculate the difference and to compare with (μ, σ) from the server side. If the difference from the sensor side lies in the range $\mu \pm \lambda\sigma$ ($\lambda = 2.5$, i.e. 98.8% confidence interval), we consider the sensor readings match with this Magnetic Trend feature.

Feature	Type	Δt_1	Δt_2	Content	Total
Moving Or Not	3	5	-	1	9
Relative Rotation			5	18	31
Magnetic Trend			5	18	31
Wi-Fi Fingerprint			-	~75	~83

TABLE I
PAYLOAD COST OF EACH FEATURE IN TERMS OF NUMBER OF BITS.

Wi-Fi Fingerprint: In Wi-Fi Fingerprint map, each location in the area contains a series of Wi-Fi signal strength readings of N_w (preset to 15) different MACs, i.e. *Wi-Fi Fingerprint*, as well as a distinguishable region. For a specific location (defined as *reference position*), its *distinguishable region* represents the locations with Wi-Fi fingerprints that have large Euclidean distances from the Wi-Fi fingerprint of the reference position. Based on the tracking results at a certain timestamp, if a target user is tracked at a reference position with a valid term in Wi-Fi Fingerprint map while some other users are in the distinguishable region, Wi-Fi Fingerprint can be extracted as a candidate feature to distinguish the target. We carefully selected N_w Wi-Fi's that are stable in each block and have the most distinguishability among different locations. Please refer to Section III-F for more details.

Figure 6 shows an example of a Wi-Fi Fingerprint map. Suppose that user 1, at the reference position, is our target to send a message. User 1 can be distinguished from user 2 using Wi-Fi Fingerprint since user 2 presents in the distinguishable region of the reference position, while user 1 cannot be distinguished from user 3. Note that the distinguishable region is around 5 meters away from the reference position, which is larger than the resolution of some existing Wi-Fi based localization scheme. This is to keep Wi-Fi Fingerprint noise-robust and ensure its reliability.

3) Context Encoding: To minimize the payload overhead, each feature is compressed into a bit string. The encoded feature structure and corresponding payload cost in bits are shown in Table I. 3 bits represent the feature *type*. 5 bits represent each timestamp (Δt_1 or Δt_2), which is used by the client to search for corresponding sensor readings. Either one or two timestamps are needed, depending on whether the feature contains an absolute or relative value. The *content* length varies among different features. *Moving Or Not* needs 1 bit to represent two states. *Relative Rotation* needs 9 bits to represent an angle (0 - 360°). *Magnetic Trend* uses 18 bits – 9 bits for μ and σ each. *Wi-Fi Fingerprint* uses, on average, 75 bits to specify 15 Wi-Fi signal strength values with different MAC addresses. Note that this is not a fixed cost since we encode the Wi-Fi signal strengths using a variant of

	Feature	User 2	User 3	User 4	Cost
✓	$f^{Moving\ Or\ Not}(\overrightarrow{t_1})$	1	0	0	9
✓	$f^{Magnetic\ Trend}(\overrightarrow{t_2})$	0	1	1	31
	$f^{Relative\ Rotation}(\overrightarrow{t_1})$	0	0	1	31
	$f^{WiFi\ Fingerprint}(\overrightarrow{t_3})$	1	1	0	46
✓	$f^{WiFi\ Fingerprint}(\overrightarrow{t_4})$	0	1	0	83
	:	:	:	:	:

TABLE II
EXAMPLE FEATURE TABLE T FOR TARGET USER 1 AND PAYLOAD COST FOR EACH FEATURE. SELECTED FEATURES MARKED BY TICKS.

Huffman coding [23] based on empirical frequencies. Details are omitted in the interest of space. The MAC addresses are sent only once when a user enters the covered area so the cost is not included.

C. Context Selection

Once the context features are extracted and encoded into the format specified above, the next task is to select the optimal set of features capable of distinguishing an individual from other people in the video.

Feature Table Construction: We define binary function D, where $D(f', f'') = 1$ if two features f' and f'' are different. $f'(= f^{type_k}_{person_i}(\overrightarrow{t}))$ and $f''(= f^{type_k}_{person_j}(\overrightarrow{t}))$ are features with the same type and timestamps, but for different people. We use the same comparison in Section III-B to evaluate D.

Based on D, we build a feature table T for each target that we want to send messages to. T is of size $m \times n$, where each row is for one feature with a certain type and some timestamps, and each column is for a person besides the target. Since in our experiment (Section IV-A), we use the features from last 30 seconds and distinguish among ten users, a typical size of T is 876×9. Each entry T_{ij} is 1 only if the jth user can be discriminated from the target by using the ith feature. Each feature is also associated with a pre-defined payload cost, C_i. An example feature table T is shown in Table II. $T_{11} = 1$ means that user 2 can be distinguished from user 1 by *Moving Or Not* at time $\overrightarrow{t_1}$.

To determine the value for T_{ij}, first we need to check whether the features f^i for both the target and the other user is *valid*. For a motion feature, we follow the criteria in [7]. For an ambience feature, we define "valid" as that a corresponding value can be found for both users from the map.

Secondly, due to time delay inherited from video streaming and packet propagation, a recorded timestamp may shift by a small amount from its actual value. We want to consider only those features which are *stable* for the target across a shift period Δs ($= 0.5s$). Namely, a feature f^i at time $\overrightarrow{t} = (\Delta t_1, \Delta t_2)$, is considered stable for T if

$$\forall s \in [0, \Delta s],\ D(f^i_{target}(\overrightarrow{t}), f^i_{target}(\overrightarrow{t+s})) = 0. \quad (1)$$

Finally, we also consider the time shift when comparing the target and other users. A feature is discriminative between the target and user j when

$$\forall s \in [0, \Delta s],\ D(f^i_{target}(\overrightarrow{t}), f^i_j(\overrightarrow{t+s})) = 1. \quad (2)$$

As a result, $T_{ij} = 1$ only if all the above three conditions are satisfied. All features in T compose a set F ($= \{f^1, \ldots, f^m\}$).

Selection Algorithm: To select the most effective features, a naive way is to decreasingly sort all candidate features according to their distinguishability/cost ratio, $H_i = \sum_j T_{ij}/C_i$. Then the features are chosen in this order until the total cost reaches a limit C_m (= 40 bytes). However, this method may fail when a specific user cannot be discriminated from the target by these selected features, even if $\sum_i H_i$ is large.

Algorithm 1: Context Selection

```
Initial selected feature set, I ← {}
Initial distinguishing power, P ← 0⃗
Number of iterations, n_iter ← 0
while n_iter < n_max do
  for each feature k ∈ I ∪ {∅}
    for each feature l ∈ Ī ∪ {∅}
      I' ← (I \ k) ∪ l
      P' ← sort(Σ_{i∈I'} T_i.)
      if P lexicographically smaller than P' and Σ_{i∈I'} C_i ≤ C_m
        I'' ← I', P ← P'
  I ← I'', n_iter ← n_iter + 1
```

Therefore, we define *distinguishing power vector* P as the sorted sum of selected rows in feature table T, and lexicographically maximize P in a greedy manner under the limit of total payload cost. This is formulated as:

$$\max P = sort(\sum_{i \in I} T_{i\cdot}),\ \text{s.t.} \sum_{i \in I} C_i \le C_m, \quad (3)$$

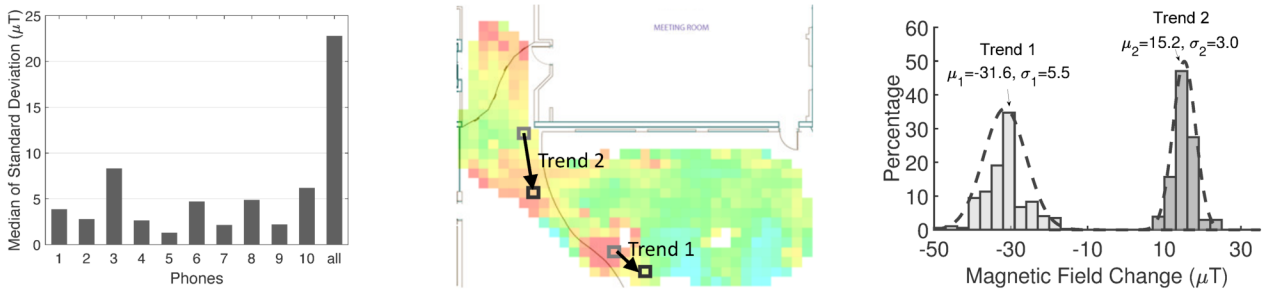
where $sort()$ ascendingly sorts the elements of a vector, and $T_{i\cdot}$ is the i-th row of matrix T. $I \subseteq F$ is selected feature set.

Lexicographical maximization of this sorted vector P guarantees that we have high distinguishability even for the least distinguishable user j, where j is the index of the smallest element in $\sum_{i \in I} T_{i\cdot}$. We can successfully send the packet only when the *normalized distinguishing power* $\hat{P} = P_1/|I|$ (note that P is already sorted and P_1 is the first element in P) is above a threshold (0.1). Otherwise, the attempt of sending the packet fails. A *sending ratio* is defined as the number of packets successfully sent over the total number of attempts.

We formulate a local search strategy (Algorithm 1) to solve this computationally hard optimization problem. It begins with an empty set and keeps applying local changes to the selected feature set I by adding, removing or substituting one feature at a time. The iteration stops when n_{iter} reaches a predefined limit n_{max}. For each iteration, we greedily maximize the increase of P by enumeration. This converged set I is used as the context address for the target user.

D. Packet Encapsulation

In the context address header, some other fields are required. The header also includes the normalized distinguishing power $\hat{P}$ (7 bits) as a threshold for context matching on the client side. Moreover, depending on the completeness of stored maps and recent locations of the target user, the server occasionally requests the target to voluntarily upload its magnetometer data and/or scanned Wi-Fi signal strengths. The solicitation for this data uses 2 bits to convey whether each type is needed. Alongside this, the transaction ID of this request (8 bits) is used to keep track of the sensor data received from the users later on. The context address header, containing all the above



(a) (b) (c)

Fig. 7. Magnetic Trend map. (a) Median of standard deviations of projected magnetometer readings among all blocks. (b) Two pairs of locations with different Magnetic Trends. (c) Normal distributions representing the two Magnetic Trends labeled in (b).

fields as well as the selected context features, is organized and encoded. It is put into the application layer of a packet along with an application message (as shown in Figure 3). The server then broadcasts the packet to all the users in the area.

E. Packet Processing on the Client Side

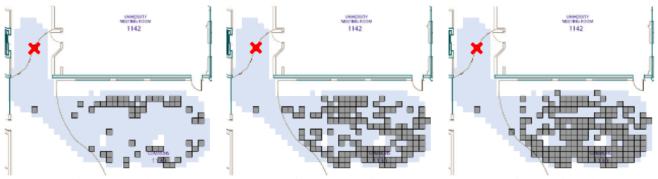
Upon receiving a broadcast packet, the user's phone decodes the context address header and extract all fields described in Section III-B3 and Section III-D. The phone extracts its corresponding sensor data for each feature at the time computed by subtracting Δt_1 and Δt_2 from the current time on the phone. Note that we do not need to consider the packet propagation delay here since it has already been dealt with during context selection (Equation 1 and 2). Each sensor-based feature is then extracted and compared with the video-based feature as discussed in Section III-B. The matching scores are averaged over all the features to obtain the overall matching score for this packet. If the matching score is greater than $\hat{P}/2$ in the received context address header, the client accepts the message in the packet and forwards it to upper-level applications.

Upon accepting a message, the client also checks the solicitation fields in the packet and may volunteer to upload the requested sensor data, e.g. magnetometer readings or scanned Wi-Fi signal strengths.

F. Map Generation

Now we come back to the two ambient sensing maps and discuss how they are generated efficiently. The first step is to divide the area covered by the camera views into a grid of small blocks. Each block is $0.5m \times 0.5m$, which determines the spatial resolution of the maps.

Magnetic Trend map: Upon receiving voluntarily uploaded sensing data, we first project the magnetometer readings to the gravity to eliminate the influence of the phone pose. As the time and location series can be easily obtained from the visual tracking process, a straightforward way is to directly use the magnetometer reading in the gravity direction as a fingerprint. The problem is that phone models and sensor quality may affect the absolute sensor readings, which is a non-negligible source of errors. Figure 7 (a) shows the median of standard deviations of projected magnetometer readings among all blocks. We observe that in the same block, the projected readings collected from the same user are consistent while the entire dataset from all users lies scattered. It inspires



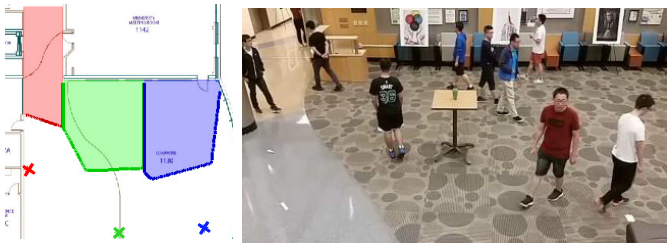
(a) After 2 minutes. (b) After 4 minutes. (c) After 6 minutes.

Fig. 8. Wi-Fi Fingerprint maps for a certain block (marked by red cross).

that, if we use a relative trend between the magnetometer readings from two different blocks collected by the same device, it is more stable than using the absolute values from different devices.

Therefore, we compute the difference between the projected magnetometer readings from one user and add it to the map, only when the user walks from one block to another. For each pair of blocks, we approximate these differences into a normal distribution, which is represented by its mean (μ) and standard deviation (σ). In Figure 7 (b), there are two pairs of blocks, whose Magnetic Trends are labeled as trend 1 and trend 2, respectively. Figure 7 (c) shows the normal distributions representing these two Magnetic Trends, where the two 98.8% confidence intervals do not overlap. Therefore, when using Magnetic Trend features as described in Section III-B2, a person who passes the block pair of trend 1 can be distinguished with high confidence, from another person who passes the block pair of trend 2 at the same timestamps. Using a distribution instead of a single value to represent Magnetic Trend provides tolerance to possible fluctuations caused by sensor noises.

Wi-Fi Fingerprint map: Wi-Fi Fingerprint map is generated through a multi-step process. Similar to Magnetic Trend, we may receive scanned Wi-Fi signal strengths from volunteer users. First, for each MAC address and each block, we compute the median of the Wi-Fi signal strengths from all users. Secondly, we calculate the variance of the medians, var, for each MAC address. var represents how the Wi-Fi signal strength differs across the blocks. Thirdly, the MAC addresses are then sorted decreasingly by var and the top N_w ($= 15$) MACs are selected for higher distinguishability among different locations. Finally, Wi-Fi Fingerprint map is generated for all blocks in the grid. It stores the medians of the Wi-Fi signal strengths of the selected MACs. All other blocks with large Euclidean distances to the *reference position* (defined in Section III-B), i.e. over a threshold of 10 dB, are marked as



(a) (b)

Fig. 9. Experiment scenario. (a) Areas covered by three cameras and corresponding camera positions. (b) An example frame in a mimic gallery.

Stage	Time in Seconds
Tracking	1.5
Context extraction	1.3
Context selection and packaging	0.2
Client-side processing	0.2
Total	**3.2**

TABLE III
END-TO-END COMPUTATION TIME.

the distinguishable region of the reference position.

Figure 8 shows an example of a Wi-Fi Fingerprint map for a reference position (marked by a red cross) at 2, 4 and 6 minutes. The walkable region is shown by light shade and the distinguishable region by dark shade. With time, as more and more data is contributed by users, the distinguishable region grows, showing that the difference between the reference position and the other blocks becomes clearer. Thus, if a target user is tracked located at the reference position in this map while another user is in the distinguishable region, this Wi-Fi Fingerprint feature can be selected to distinguish these two users.

In real cases, the server does not need to frequently send solicitation requests. Once the collected dataset is large enough, the server holds back on solicitation for that area. An expiration time can also be set to void outdated sensing data. In our experiments, we ignore these two factors due to the short experiment period.

IV. EVALUATION

A. Experimental Setup

In our real-world experiment, three Samsung Galaxy S5 smartphones are used as IP cameras to capture and stream videos at a frame rate of 13 fps, a bit rate of 2000 kbps, and a resolution of 800×480. We set up our server on three PCs with dual NVIDIA GTX 1080 Ti SLI, and run MATLAB and C++ programs on each. A software called ClockSynchro [24] is used to synchronize these computers. Google Pixel XL smartphones are employed as clients, which log accelerometer, gyroscope, gravity, magnetometer readings, and Wi-Fi scan results at $400Hz$, $400Hz$, $200Hz$, $50Hz$ and $1Hz$ respectively. They also run our Android client app to receive packets.

We evaluated our system in a real-world scenario of an "art gallery" in a university lobby with a walkable area of $107m^2$. The area covered by each camera is shown in Figure 9(a) with shades and the camera positions are marked with crosses. We invited 10 volunteers to naturally walk around or stop by

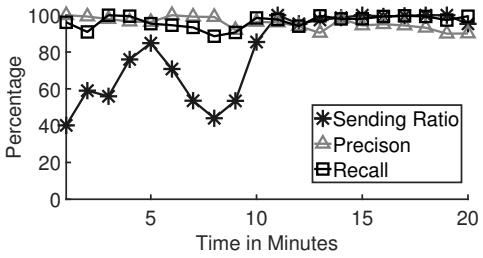


Fig. 10. Overall precision, recall and sending ratio of the system.

at the paintings as they pleased, with a smartphone put in their pockets. Figure 9(b) shows an example frame from one camera, with 10 users in the gallery. We tested the utility of the context address by sending messages to all 10 users every two seconds, i.e. 6000 messages sent in a 20-minute session.

B. Performance Results

We intend to concentrate on the following aspects:

(1) System Overall Performance

We evaluate the overall precision and recall rate of our system. The precision represents the ratio of the messages accepted by a user which are actually targeted for it. The recall is the ratio of the messages targeted for a specific user which are successfully accepted by it. Figure 10 shows that, as magnetometer and Wi-Fi readings gradually contribute to the map generating process, the performance starts to improve after about 10 minutes. After the cold start period, the average precision of our system is 93.4% throughout the last 10 minutes while the recall rate is 98.3%. Moreover, the sending ratio (defined in Section III-C) increases sharply and reaches an average of 98.5%. The combination of motion and ambience features leads to an overall high precision and recall rate, showing that the context features used have a high distinguishability and the maps are stable over time.

Table III shows the median of computation time for different processing stages through 20 minutes. The total computation time is 3.2 seconds, in which the tracking process takes the largest portion, i.e. 1.5 seconds. The tracking time could be shrunk if a faster and more accurate tracking scheme can be introduced into our system. And other stages, i.e. context extraction, context selection and packaging, and client side processing, take 1.3, 0.2 and 0.2 seconds respectively. This demonstrates the efficiency of our context selection algorithm.

(2) Packet Overhead

We set the maximum number of bytes for the context address header to 20, 40 and 100, and evaluate how it affects the performance during the last 10 minutes. Figure 11 shows the results. When the limit is set to 40 bytes (i.e. same as an IPv6 header), it already achieves similar performance as it's extended to 100 bytes. In PHADE [7], the packet overhead is a severe problem since the large coefficient matrix needs to be sent out, no matter how many users that the system is trying to communicate with. For example, same as in our experiment, if PHADE is sending messages to 10 users at the same time, the average packet overhead for each user is 200 floats, i.e.

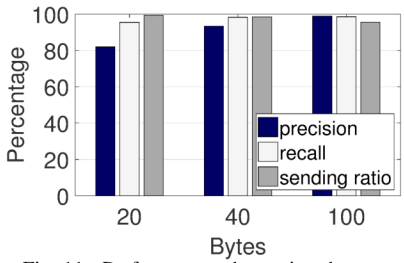


Fig. 11. Performance when using the context address header of various maximum length.

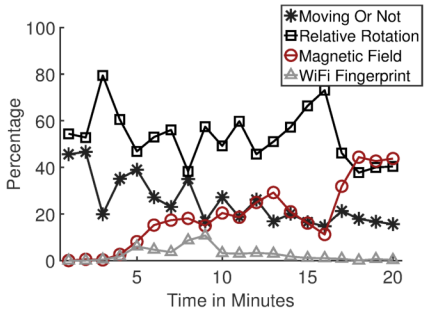


Fig. 12. The proportion of different types of features selected for the context address.

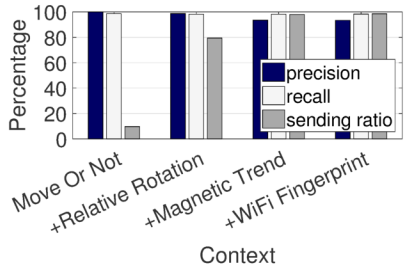


Fig. 13. Performance by adding different types of context features.

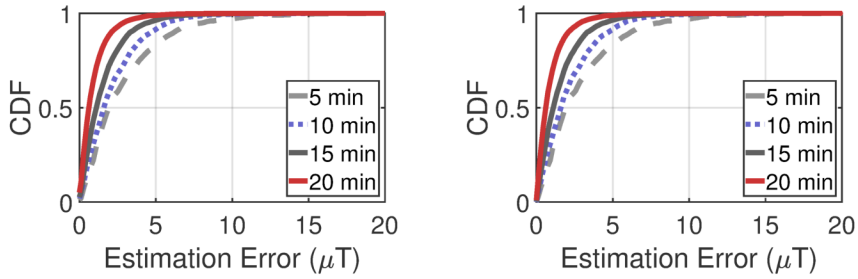


Fig. 14. Error in the difference of magnetometer readings over time. (a) and (b) show the distribution of errors in estimated mean and standard deviation, respectively.

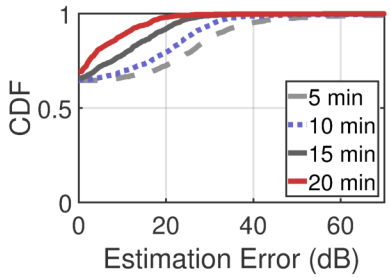


Fig. 15. Error in Wi-Fi Fingerprint map over time.

800 bytes. In contrast, the packet overhead in our system is always less than 40 bytes regardless of the number of users, only 5% of PHADE.

(3) Selected Context Features

The types of context features selected over time are shown in Figure 12. During the initial period, only *Move or Not* and *Relative Rotation* are selected. This is because the system was at a start-up stage, waiting for users to voluntarily upload their sensor data. With more and more valid sensor data contributed to building the ambient sensing maps, *Magnetic Trend* and *Wi-Fi Fingerprint* features start to be selected. The percentage of *Magnetic Trend* increases greatly after the first 4 minutes. The *Wi-Fi Fingerprint* has limited contribution because its payload cost is larger than the other three types.

Figure 13 shows how the system performance changes when adding different types of context features. When adding more types of features, the sending ratio is largely enhanced while the precision and the recall remain high.

(4) Maps Generated Over Time

To build the ground truth maps, we use the ground truth messaging destinations to get all available sensor data from the users. We evaluate Magnetic Trend map in terms of errors of the mean and the standard deviation of the difference in each pair of blocks. Figure 14 shows that as there are more and more sensor data, these two kinds of errors gradually and steadily decrease over time. After about 10 minutes, the median of both errors drops below $1.5\mu T$. This implies that the map generation converges in a short time. Similarly, we evaluate the errors of Wi-Fi Fingerprint map by calculating the distances from the medians of Wi-Fi signal strengths. Figure 15 shows that with more Wi-Fi readings uploaded, the Wi-Fi Fingerprint map gradually approaches the ground truth.

C. Video Simulation

We also want to gain insight into the scalability and feasibility of our system in a retailer scenario with dense people. Since we want to observe the natural behavior of human, we run an alternative simulation to approximate our system's performance when deployed in a real retail store. Instead of obtaining real sensor readings from the smartphones carried by people, we synthesize sensor features by injecting statistical noises into video-based features, to simulate the inconsistency between the video and the sensor sides.

To conduct the simulation, we first record videos in a retail store during its busy hours at 5-6pm. Two cameras cooperate to cover a walkable area of $105m^2$ and each video is 6 minutes long. The cameras capture up to 13 people simultaneously and the entire videos include 52 distinct people. Figure 16 is an example frame illustrating the typical people density in the video. We then conduct pedestrian tracking on these videos and extracted the video-based context features for each person. To synthesize sensor-based motion features, we analyze the error distributions using the data collected in Section IV-A and inject noises based on these distributions into the video-based features. For synthesizing sensor-based ambience features, we wardrive this area for 20 minutes, and use this data for both building the maps and fitting the error distributions for magnetometer and Wi-Fi data. The wardriving is also conducted between 5-6pm to ensure the collected Wi-Fi signal strengths reflect the actual readings with crowds in the surrounding. Finally, we simulate the message sending process and each person accepts messages by comparing the synthesized sensor features with the actual video-based features.

The simulation demonstrates that our system can distinguish a person in practical and dense scenarios, reaching a sending ratio of 90.0%, a precision of 99.7% and a recall of 95.3%.

Fig. 16. Video simulation scenario.

V. Discussion

Limitations: As our system highly depends on the performance of pedestrian detection and tracking schemes, it may not work well in scenarios with lots of obstacles in the environment and human mutual occlusion. If the tracking fails occasionally, the user is treated as a new person just entering the area. Therefore, the chance of successfully delivering the messages is affected.

Other possible features: Since our system has explored multiple ways to represent features (e.g. single data point, trend, and fingerprint), it can also be extended to use other features, such as light intensity, walking direction, step phase, Bluetooth, 5G signals, etc. They may contribute differently in various use cases, for example, step phase can be used to distinguish people walking along a similar trajectory.

Difference with indoor localization: One may wonder if we built another indoor localization system. The short answer is "No". Our system, as a general framework for direct camera-to-human communication, can be adapted into more various application scenarios, as discussed in Section I, including indoor localization. Some systems with similar communication purposes [7] have demonstrated these applications with real-world evaluations. Even if we had a perfect indoor localization system, this location information is not suitable for a communication system like ours. One main reason is the potential privacy leakage from broadcasting location information.

Broadcast methodology: In our system implementation, we use Wi-Fi as the broadcast media. But there are also other options, such as LTE Direct, BLE advertisement, etc.

VI. Related Work

Message delivery based on visual tracking. Recent works have built some communication paths to send messages to a targeted person in surveillance camera views. PHADE [7] uses people's walking behaviors as their temporary communication addresses. However, in some crowded scenarios, merely using motion features does not provide enough distinguishability to represent each person. Also, PHADE transmits large coefficient matrices along with address codes, which introduces a non-negligible packet overhead. On the other hand, our system utilizes both motion and ambience features to obtain higher distinguishing ability and introduces a context address header with a small fixed length while providing accurate message targeting. Another work, TAR [8] uses a combination of multi-camera human tracking and Bluetooth proximity sensing to conduct ID association and deliver targeted advertisements. When some people are in close proximity, TAR needs BLE readings for a longer period to identify a person among them. However, the ability to discriminate a user from its companions is insufficient since Bluetooth proximity is the only feature used. Our system can distinguish a person from a denser group relying on richer context features, even if some people locate closely or behave similarly. Also, it requires Bluetooth on all users' smartphones to be on and continuously broadcast BLE signals, which raises severe privacy concerns.

Human ID association. Existing schemes for human ID association use various techniques and devices for identification. [5] has used the accelerometer readings from a sensor worn on a person's belt to develop an ID matching algorithm for associating people. Another work, Insight [6], uses the motion patterns and clothing colors to recognize people. These patterns serve as a temporary fingerprint for an individual. Both of these schemes depend on the users to upload their sensor data while we can still correctly identify a user even if it does not upload any data. Also, in contrast to [5], [6], we have implemented our idea into a real-time end-to-end system. Among other approaches, ID-Match [4] can recognize and correctly assign IDs to individuals using relative motion paths of RFID tags worn by people and 3D camera. For outdoor environments, RFID and BLE are combined with a stereo-based identification system in [25]. In these approaches, the identification relies on users wearing RFID tags or BLE beacons. It is hard to ensure everyone carries its tag in a large public area, hence rendering these schemes infeasible for such environments. Our system associates a user in the camera view with its smartphone without requiring tags or preregistration.

Camera sensor combination. Research based on combining cameras and sensors has been popular in recent past with widespread applications. Gabriel [26] uses image capturing and mobile sensing to develop a cognitive assistance system. Smartphone's motion and light sensors combined with cameras allow authors in [27] to enhance the biometric authentication process through facial recognition. Overlay [28] uses a fusion of a smartphone camera and sensors to enable augmented reality on the phone via building a geometric representation of the environment. We introduce the novel concept of using cameras and smartphone sensors to allow communication between the camera and people in the camera view with applications in public safety and other day-to-day activities.

VII. Acknowledgments

We sincerely thank the anonymous reviewers for their insightful suggestions and comments, and Purdue Research Foundation for partially funding this research.

VIII. Conclusion

This paper solves the problem of digitally associating people in a camera view to their smartphones without knowing their IP/MAC addresses. A fully operational real-time prototype system is developed, which utilizes a context address consisting of motion patterns and ambience to identify each person. We deploy an efficient context selection algorithm to choose discriminative features and fit them into a fixed-length header. We also generate ambient sensing maps in an effortless way. Our system achieves a sending ratio of 98.5%, an acceptance precision of 93.4%, and a recall of 98.3%.

REFERENCES

[1] N. Jenkins, "245 million video surveillance cameras installed globally in 2014," *IHS Technology*.

[2] K. W. Bowyer, "Face recognition technology: security versus privacy," *IEEE Technology and society magazine*, vol. 23, no. 1, pp. 9–19, 2004.

[3] "San Francisco Banned Facial Recognition. Will California Follow?" https://www.nytimes.com/2019/07/01/us/facial-recognition-san-francisco.html.

[4] H. Li, P. Zhang, S. Al Moubayed, S. N. Patel, and A. P. Sample, "Id-match: A hybrid computer vision and rfid system for recognizing individuals in groups," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: ACM, 2016, pp. 4933–4944. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858209

[5] D. Jung, T. Teixeira, and A. Savvides, "Towards cooperative localization of wearable sensors using accelerometers and cameras," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[6] H. Wang, X. Bao, R. R. Choudhury, and S. Nelakuditi, "Insight: recognizing humans without face recognition," in *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*. ACM, 2013, p. 7.

[7] S. Cao and H. Wang, "Enabling public cameras to talk to the public," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 2, pp. 63:1–63:20, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3214266

[8] X. Liu, Y. Jiang, P. Jain, and K.-H. Kim, "Tar: Enabling fine-grained targeted advertising in retail stores," in *MobiSys*, 2018.

[9] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *ECCV*, 2012, pp. 645–659.

[10] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[11] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.

[12] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[13] A. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 666–673.

[14] J. Xing, H. Ai, and S. Lao, "Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1200–1207.

[15] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1265–1272.

[16] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1201–1208.

[17] B. Yang and R. Nevatia, "Online learned discriminative part-based appearance models for multi-human tracking," in *European Conference on Computer Vision*. Springer, 2012, pp. 484–498.

[18] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1323–1330.

[19] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1313–1320.

[20] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1285–1292.

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

[22] N. Peterfreund, "Robust tracking of position and velocity with kalman snakes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 21, no. 6, pp. 564–569, 1999.

[23] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[24] "Time synchronization in a local network," http://clocksynchro.com/.

[25] D. F. Llorca, R. Quintero, I. Parra, and M. A. Sotelo, "Recognizing individuals in groups in outdoor environments combining stereo vision, rfid and ble," *Cluster Computing*, vol. 20, no. 1, pp. 769–779, Mar. 2017. [Online]. Available: https://doi.org/10.1007/s10586-017-0764-0

[26] G. Takacs, V. Chandrasekhar, and Others, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *ACM ICMR*, 2008.

[27] S. Chen, A. Pande, and P. Mohapatra, "Sensor-assisted facial recognition: An enhanced biometric authentication system for smartphones," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. New York, NY, USA: ACM, 2014, pp. 109–122. [Online]. Available: http://doi.acm.org/10.1145/2594368.2594373

[28] P. Jain, J. Manweiler, and R. Roy Choudhury, "Overlay: Practical mobile augmented reality," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2015, pp. 331–344.