FusionTalk: An IoT-based Reconfigurable Object Identification System

Ling-Yan Zhang*, Hung-Cheng Lin, Kun-Ru Wu, Yi-Bing Lin, Fellow, IEEE, and Yu-Chee Tseng, Fellow, IEEE

Abstract—Multi-sensor data fusion combines various information sources to produce a more accurate or complete description of the environment. This paper studies an object identification (OID) system using multiple distributed cameras and Internet of Things (IoT) devices for better visualizability and reconfigurability. We first propose a data processing and fusing method to merge the detection results of different IoT devices and video cameras, in order to locate, identify, and track target objects in the monitored area. Then, we develop the FusionTalk system by integrating the data fusion techniques with IoTtalk, an IoT device management platform. FusionTalk is designed with flexibility, modularity, and expansibility, where cameras, IoT devices, and network applications are modularized and can be conveniently plugged in/out, reconfigured, and reused through graphical user interfaces. In FusionTalk, the scope and the target of surveillance can be flexibly configured and associated, and administrators can be warned and easily visualize the movement and behavior of specific objects. Our experimental evaluation of the data fusion algorithm in various scenarios shows an identification accuracy above 95%. Finally, theoretical and numerical analyses on the failure probability of pairing IoT devices with video objects by FusionTalk are presented. Extensive experiments are performed to demonstrate the pairing effectiveness in real-world scenarios with failure probability less than 0.01%.

Index Terms—data fusion, IoT, object identification (OID), tracking, video surveillance.

I. INTRODUCTION

OBJECTIVE identification (OID) is a long existing problem in many security, surveillance, and facility management applications. Its tracking targets include buildings, people, vehicles, valuable things, etc. [1]. Traditional video surveillance systems [2], [3] normally track moving objects using cameras. They are susceptible to changing environment factors and target appearances. By identification, we intend to know not only object types (human, dog, item type, etc.) but also their unique identities (name, sequence number, etc.).

We see a new opportunity offered by the widely accepted IoT devices, such as sensors, smart watches, smart phones, badges, pet chips, and RFIDs. These devices have become virtually the unique identity of an object. Such IoT devices are equipped with some inertial sensors and, more importantly, have access to the owners’ profiles that may reveal a lot of critical information (such as identity, age, sex, citizenship, social network, past activity, purchase record, etc.). Clearly, using IoT devices can provide some complementary information to what is offered by cameras. However, interacting with distributed IoT devices needs significant efforts, not to mention how to measure, fuse, and associate the data coming from various devices with different sample rates, transmit delays, and data representations.

We design an OID system named FusionTalk to handle the aforementioned multi-sensory data fusion issue [4], [5], [6], [7] under an IoT device management platform called IoTtalk [8]. FusionTalk integrates the object detection and sensor feature information (such as moving trajectory, gesture, and gait) coming from distributed cameras, IoT sensors, and wearable IoT devices. It then identifies and visualizes objects according to the pairing results of the fusion model. Different from traditional video surveillance systems [2], [3], FusionTalk allows users to specify a camera to track specific objects through a web-based graphical user interface (GUI), and it not only tracks objects but also visualizes their profiles, movements, and behaviors according to application requirements. Fig. 1 shows a smart surveillance scenario in a gallery. In this scenario, various objects, such as people, pets, and paintings, can be recognized and tagged by their profiles in a monitor screen. An anti-theft alarm of a painting stores its basic information and environment conditions such as painter, temperature, and humidity, and a pet chip pulls up all the necessary information such as name, owner, next appointment, and medical records. Our implementation and evaluation show promising results. The contributions of this work are summarized as follows.
We propose an IoT-based OID system named FusionTalk, which is developed based on a unified architecture that modularizes different devices, fusion algorithms, and network applications by emphasizing on uniform interface, connectivity, and device management, making it highly reconfigurable for various applications by reusing data from different IoT devices.

We propose a data processing and fusing method to merge the detection results of different IoT devices and video cameras, which can handle the synchronization and disruption issues of distributed sensors. With Multiple sensors, the tracking capability of FusionTalk greatly outperforms single-sensor solutions.

We implement and evaluate the system in various scenarios, which shows promising results with an identification accuracy above 95%. And we also present the theoretical derivation and numerical analyses on the failure probability of FusionTalk under different network conditions, which shows a failure probability less than 0.01%.

The remainder of the paper is organized as follows. Section II discusses the related work. Section III presents the design of FusionTalk. Section IV describes our system prototype. Section V investigates some IoT delay performance issues. Section VI concludes the paper.

II. RELATED WORK

Object identification (OID) has been extensively studied in the computer vision and IoT fields. It plays a critical role in many security, surveillance, and business-intelligence applications. Significant progresses have been made in vehicle identification [3], [9], pet monitoring [10], [11], object localization and tracking [12], [13], and video surveillance [2], [14]. In this section, we briefly review the state-of-the-art biometric-based OID approaches [11], [15], [16], [17], [18], RFID-based OID approaches [10], [12], [13], [19], and multi-sensory fusion solutions [20], [21], [22].

Biometric-based OID approaches recognize objects by measuring their unique physical or behavioral characteristics, mainly including face, hand geometry, fingerprint, and iris. Parkhi et al. [15] conducted face recognition using a deep convolutional neural network. Kumar and Singh [11] captured dog face images from cameras and recognized individual dogs based on their facial biometric features. However, face recognition solutions are sensitive to lighting condition, occlusion, and viewing direction. Rig et al. [16] analyzed a finger-vein-based biometric identification system under different image quality conditions. But, fingerprint [17] and iris recognitions [18] need users’ biology features, and require close contact to specific devices. Hence, biometric-based solutions are more suitable for indoor applications such as smart home, physiology monitoring, and behavior analysis. Applying them to large-scale OID would be difficult.

RFID-based OID approaches locate a specific RFID tag relying on RSSI and phase measurements. Li et al. [12] presented an RFID-based system for tracking random moving objects in indoor environment. Lai et al. [13] proposed an RFID-based relative localization system for detecting the order of tags. Xiao and Green [19] improved localization performance by adding one more RFID tag. Saeed et al. [10] proposed an RFID-based pet monitoring and identification system. However, RFID needs multiple cooperative RFID tags/readers to observe direction information. As RFIDs are highly sensitive to multipath effects, it is difficult to achieve high-precision localization in a moving crowd or dynamic environments.

Multi-sensory fusion solutions identify and track objects by fusing data from multiple sensors and devices. Theoretically, the best identification and tracking performance can be achieved by implanted or wearable devices because they are personalized as long as they are trackable. On the other hand, cameras work very well with visualization effect as long as object image quality is satisfactory, no occlusion happens, and dataset is well collected. Therefore, there is a great opportunity to enhance OID if we can conduct fusion with IoT devices and cameras. Recently, there has been much effort on employing multi-sensor data fusion for human localization [20], [21], [22], activity recognition [23], people identification [24], [25], and smart ubiquitous environments [26]. Jain et al. [14] presented an automated surveillance system by using a Raspberry Pi with passive infrared sensors for motion detection and a remote camera for video recording. Minh et al. [20] fused data from passive infrared (PIR) sensors and wearable inertial measurement units (IMUs) to estimate locations of a single resident, but cannot track multiple residents living together. Dezhengi et al. [23] investigated data fusion from multiple IMUs to improve the gait identification generalization performance. Xiong et al. [24] extracted thermal infrared features from different human parts through multiple PIR sensors for human identification. Han et al. [25] identified occupants based on their interaction behaviors with objects via IMUs. A literature survey of data fusion for IoT with a particular focus on mathematical methods is in [26].

Our OID system aims to combine information from multiple IoT devices to improve accuracies, inferences, and visualization effect. We consider IoT devices including sensors, wearable devices, and video cameras. Such an OID system may take advantage of the characteristics of different devices, as pointed out in [4], [7]. However, integrating distributed IoT devices poses a big challenge, especially when they work in different sample rates, transmission delays, and forms of data representation. For example, the surveillance systems [2], [14] have a static and centralized structure where the fusion server directly connects to terminal devices. As IoT industry progresses very fast, an OID system should be highly distributed through modularization so that new components can be dynamically plugged in/out, and their data/results can be flexibly reconfigured and reused for changing application requirements. Recently, a platform called IoTtalk [8] has been developed to ease the management of distributed versatile IoT devices [27], [28], [29], which motivates us to design the proposed FusionTalk solution. Different from the current state of the art, the FusionTalk system addresses the asynchronous time and the warped data problems of distributed sensors, and provides high identification accuracy and low failure probability even in dynamic environments.
TABLE I
SUMMARY OF NOTATIONS

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$DF$</td>
<td>device feature</td>
</tr>
<tr>
<td>$IDF, ODF$</td>
<td>input device feature and output device feature</td>
</tr>
<tr>
<td>$SA, DA$</td>
<td>sensor application and device application</td>
</tr>
<tr>
<td>$o_i$</td>
<td>the $i$-th object detected from the videos</td>
</tr>
<tr>
<td>$w_j$</td>
<td>the $j$-th wearable device detected from the environment</td>
</tr>
<tr>
<td>$V_k^t, S_k^t$</td>
<td>the number of features; the feature set of $o_i$ and $w_j$ over a period of time $t$ for the $k$-th feature, $k \leq K$</td>
</tr>
<tr>
<td>$V_k^t(k), S_k^t(k)$</td>
<td>the data set of $V_k^t$ and $S_k^t$ for the $k$-th feature</td>
</tr>
<tr>
<td>$M, N$</td>
<td>the number of data points of $V_k^t(k)$ and $S_k^t(k)$</td>
</tr>
<tr>
<td>$Sim(V_k^t, S_k^t)$</td>
<td>the similarity score of $V_k^t$ and $S_k^t$</td>
</tr>
<tr>
<td>$w_1\star$</td>
<td>the best matched wearable device for $o_1$</td>
</tr>
<tr>
<td>$msg_1$</td>
<td>the message sent from the ObjLoc to DFusion server</td>
</tr>
<tr>
<td>$msg_2$</td>
<td>the message sent from the ObjTraj to DFusion server</td>
</tr>
<tr>
<td>$t_1, t_2$</td>
<td>the delay of message $msg_1$ and $msg_2$</td>
</tr>
<tr>
<td>$E[t_1], E[t_2]$</td>
<td>the expected value of $t_1$ and $t_2$</td>
</tr>
<tr>
<td>$V[t_1], V[t_2]$</td>
<td>the variance of $t_1$ and $t_2$</td>
</tr>
<tr>
<td>$t_0$</td>
<td>the delay difference between $t_1$ and $t_2$</td>
</tr>
<tr>
<td>$t_3$</td>
<td>the threshold of delay difference</td>
</tr>
<tr>
<td>$Pr[t_3 &lt; t_0]$</td>
<td>the failure probability of FusionTalk</td>
</tr>
<tr>
<td>$f_k(t_k)$</td>
<td>the Erlang density function, $k = 1, 2$</td>
</tr>
<tr>
<td>$\lambda_k$</td>
<td>the shape parameter</td>
</tr>
<tr>
<td>$\lambda_v$</td>
<td>the rate parameter</td>
</tr>
<tr>
<td>$1/\beta$</td>
<td>the mean of $t_3$, which is exponentially distributed</td>
</tr>
</tbody>
</table>

III. DESIGN OF FUSIONTALK

FusionTalk is developed based on the device management platform IoTtalk [8]. In IoTtalk, any physical IoT device and cyber IoT application (such as web application, AR/VR animation, and mobile app) can be characterized by its functionality named device feature (DF), which is a specific input or output “capability” of an IoT device. For example, a smartphone with a gyro sensor has an input device feature (IDF) called “direction”. A computer with a monitor has an output device feature (ODF) called “display”. An IDF can be connected to an ODF through a “join” link. A function can be embedded in a join link to convert input data (e.g., temperature) to output data (e.g., on-off status). In the IoTtalk GUI, a user can click on join links to edit functions that manipulate the data from IDF(s) to ODF(s). IoT devices can be connected to an IoTtalk server through the Internet. Whenever the data of an IDF from its input device are updated, the associated ODFs will be triggered to affect the corresponding output devices. For convenience, we summarize the major notations in Table I.

Fig. 2 shows the FusionTalk architecture, which contains device domain and network domain. In the device domain, every IoT device consists of a sensor/actuator application (SA) and a device application (DA). The SA implements the sensor/actuator logic including data acquisition and processing, and specifies the IoT message format. The DA communicates with the IoTtalk server for device registration and data exchange between the SA and the IoTtalk server through IDF(s) (Fig. 2(1)) or ODFs (Fig. 2(2)). Note that an IoT device may contain both IDF(s) and ODF(s). In the network domain (Fig. 2(3)), the execution and control system (EC) defines the HTTP based RESTful or the MQTT APIs to deliver/retrieve the corresponding IDF/ODF information. These IoT data are stored in a SQL database (DB). The GUI provides a friendly web-based interface (Fig. 3) for the users to quickly establish the desired connections and interactions among distributed IoT devices. Through the GUI, one can easily develop FusionTalk to set up device modules, DFs, connection configurations, and desired tasks, thus providing a convenient and flexible platform for connecting various IoT devices across different networks and easily sharing IoT devices and data across applications [27], [28], [29].

An example of applying FusionTalk to IoT devices in two stories is in Fig. 3. There are four types of physical input devices: ObjTraj (video camera for trajectory tracking; see Fig. 3(1)), ObjPose (video camera for pose tracking; see Fig. 3(3)), ObjLoc (beacon and wearable for location tracking; see Fig. 3(5)), and ObjMot wearable for motion tracking; see Fig. 3(7)). A cyber input device Cmd (Fig. 3(9)) is developed for issuing commands input and visualizing results. FusionTalk automatically creates a software module for every server, which is represented as an icon in the GUI. Fig. 3(2) is the icon for the software module of the ObjTraj server. Similarly, Fig. 3(4), (6), (8), and (10) are the icons for the ObjPose, ObjLoc, ObjMot, and Cmd servers, respectively. These input devices are characterized by the following IDFs: Traj-I (Fig.
3(2) for trajectory descriptions of target objects calculated from videos), Pose-I (Fig. 3(4) for pose descriptions of target objects calculated from videos), Loc-I (Fig. 3(6) for location descriptions of target objects calculated from beacons and wearables), Mot-I (Fig. 3(8) for motion descriptions of target objects calculated from wearable devices), and Cmd-I (Fig. 3(10) for administration commands on target objects).

FusionTalk has four cyber output devices called DFusion, which are servers for pairing video data with different IoT devices (see Fig. 3(11)-(18) for these servers and their software module icons). DFusion has five ODFs, namely Traj-O, Pose-O, Loc-O, Mot-O, and Cmd-O, and their formats remain the same as Traj-I, Pose-I, Loc-I, Mot-I, and Cmd-I, respectively. The join links in the middle define how these data are shared and combined for the output devices. Each fusion server groups the data of ODFs and then implements data fusion algorithms to associate video objects with their IoT data for the goal of OID.

The advantage of FusionTalk is that both device modules, data connections, and applications are highly reconfigurable. Firstly, once we define a device module with its IDFs, any IoT device providing the same format of data can be connected to FusionTalk by reusing the module. Secondly, through joins, the data of IDFs can be reused by multiple ODFs. Thirdly, applications can be derived independently of devices. For example, applications are only dependent on the ObjLoc module, but not on what sensors and underlying algorithms are used.

A. SA of Input Devices

The sensor application (SA) of each device is responsible for data acquisition, retrieval of useful high-order information, and transmission according to the specified formats. Fig. 4 shows the hardware components of FusionTalk. We use cameras to produce human trajectories and poses, and beacons and wearables to produce human locations and gestures. FusionTalk defines five input device modules, as shown in Fig. 3. Their SAs are detailed as follows.

ObjTraj retrieves the trajectory information of target objects from its connected cameras. A server can connect to several cameras according to its capacity. In ObjTraj, we use YOLO (You Only Look Once) v3 [30] to retrieve target objects from each video frame. Every object is marked with a bounding box, whose center is used as the object’s location. Objects’ moving trajectories are derived from their continuous locations in consecutive images. Via SORT (Simple Online and Realtime Tracking) [31], objects’ trajectories can be obtained by tracking their bounding boxes. ObjTraj has one IDF called Traj-I that represents the trajectory information of all detected objects. Traj-I has the format \([t, CamID, Box_1, Box_2, ...]\), where \(t\) is the timestamp, \(CamID\) is the camera ID, and \(Box_i = [BoxID_i, x_i, y_i, w_i, h_i, c_i]\). In \(Box_i\), \(BoxID_i\) is the bounding box ID of the \(i\)-th object, \((x_i, y_i)\) is the center, \(w_i\) is the width, \(h_i\) is the height, and \(c_i\) is the confidence score in the range \([0, 1]\).

ObjPose detects and estimates 2D poses of target objects from its connected cameras. We use OpenPose [32] to detect the 2D human poses. OpenPose can jointly detect up to 135 human body, hand, facial, and foot keypoints per person. Since we only concern about a person’s trajectory or gesture, we ignore other keypoints and only use the BODY_25 model of OpenPose, which produces 25 keypoints per human body. ObjPose has one IDF called Pos-I that represents the 2D pose information of all detected objects. Pos-I has the format \([t, CamID, Pos_1, Pos_2, ...]\), where \(t\) is the timestamp, \(CamID\) is the camera ID, and \(Pos_i = [PosID_i, \(x_{i,1}, y_{i,1}, c_{i,1}\), \(x_{i,2}, y_{i,2}, c_{i,2}\), ..., \(x_{i,25}, y_{i,25}, c_{i,25}\)]\). In \(Pos_i\), \(PosID_i\) is the pose ID of the \(i\)-th object and the rest are the coordinates and confidences of the 25 keypoints.

ObjLoc tracks target object locations through wireless localization techniques by both wearable devices and BLE beacons. BLE beacons are deployed in the surveillance region to track the Bluetooth signal of objects’ wearable devices. We design a wearable device which has a Raspberry Pi, inertial components, and a Bluetooth module, as shown in Fig. 4(c). We adopt the positioning algorithm [33] that combines gait information and beacon signal strengths by a particle filter. Fig. 5 shows an example, where 6 beacons are deployed on the ground. Therefore, we can observe two persons’ trajectories from both outputs of ObjTraj and ObjLoc simultaneously. Note that location tracking can be conducted by not only beacons, but also the fusion of inertial sensors. The details are out of the scope of this work and can be found in [34], [35]. ObjLoc has one IDF called Loc-I that represents the locations of objects in format \([t, Loc_1, Loc_2, ...]\), where \(t\) is the timestamp and \(Loc_i = [ObjID_i, x_i, y_i]\) is the location data of the \(i\)-th object. In \(Loc_i\), \(ObjID_i\) is its ID and \((x_i, y_i)\) is its location. By connecting these locations, we are able to derive an object’s moving trajectory.

ObjMot extracts motion features of target objects from their wearable IoT devices equipped with inertial sensors, such as accelerometer, gyro, and magnetometer. We use a smart phone (HTC 10) to emulate a wearable IoT device. This
B. SA of Output Device

The moving trajectories produced by cameras and beacons are similar and thus can be fused and paired as illustrated in Fig. 5. Similarly, motion features produced by cameras and wearables are similar and can be fused as illustrated in Fig. 6. The SA of DFusion takes data from cameras and sensors (i.e., beacons or wearables) and implements the fusion algorithms to produce OID results. For an object $o_i$, $V_i^t$ and $S_j^t$ are sequences of features of $o_i$ extracted over a period of time $t$. Fig. 7 shows the data fusion process, which has four steps:

Step 1. Detect all objects $\{o_1, o_2, \ldots\}$ from the videos.

Step 2(a). Retrieve the feature set $V_i^t$ of each $o_i$ from the videos.

Step 2(b). Retrieve the feature set $S_j^t$ of each wearable $w_j$ from the sensors.

Step 3. Retrieve the environment information to transform $V_i^t$ and $S_j^t$ to the same space so as to make proper comparisons.

Step 4. If there is high correlation between $V_i^t$ and $S_j^t$, argument the video object $o_i$ by wearable $w_j$’s data.

At Step 2, the extracted features include trajectory features, location features, pose features, or motion features. At Step 3, the environment information includes camera parameters, landmarks, and beacons’ localizations. At Step 4, we use the dynamic time warping (DTW) algorithm [36] to measure the similarity score between two time series, $V_i^t$ and $S_j^t$. DTW recursively examines all subsequence combinations of the whole sequences to obtain the best match, and it can handle the synchronization and disruption issues of distributed sensors.

Suppose that each of $V_i^t$ and $S_j^t$ includes $K$ sub-features and we have obtained $M$ data points from object $o_i$ and $N$ data points from wearable $w_j$ for the $k$-th feature ($k \leq K$) at time $t$. Let $D_k(m, n)$ be the DTW distance of the $k$-th feature between subsequences $V_i^t[1 : m]$ and $S_j^t[1 : n]$, $m \leq M$ and $n \leq N$. The initial condition is $D_k(1, 1) = d(V_i^t(1), S_j^t(1))$, where $d(\cdot)$ is the distance function between two data points (our study uses Euclidean distance), and $D_k(m, n)$ is defined as:

$$D_k(m, n) = \min\{D_k(m - 1, n), D_k(m, n - 1)\} + d(V_i^t(m), S_j^t(n)). \quad (1)$$

This recursive definition helps us to progress in an asynchronous manner. The goal of DTW is to find a mapping path between $V_i^t[1 : M]$ and $S_j^t[1 : N]$, such that the total distance is minimized.
distance on this mapping path is minimized. The final result is $D_k(M, N)$. To further normalize the similarity score of each feature, we define the maximum distance of $K$ features to be $D_{max}$. The similarity score of the $k$-th feature ($k \leq K$) between $V_i^t$ and $S_j^t$ is defined as:

$$Sim_k(V_i^t, S_j^t) = 1 - \frac{D_k(M, N)}{D_{max}}.$$  

Clearly, $0 \leq Sim_k(V_i^t, S_j^t) \leq 1$ and 1 means most similar. Finally, the comprehensive similarity score between $V_i^t$ and $S_j^t$ is obtained by combining similarity scores of $K$ features, written as:

$$Sim(V_i^t, S_j^t) = \sum_{k=1}^{K} \alpha_k Sim_k(V_i^t, S_j^t),$$

where $\alpha_k$ is a weight derived from experiences.

Based on these similarity scores, video objects are matched with wearable devices by a greedy approach. Specifically, for each object $o_i$, we find its best matched wearable device $w_j$, at time $t$ as follows:

$$w_j = arg\min_{all\ w_j} (Sim(V_i^t, S_j^t)).$$

Since multi-pairing should be avoided, a paired device will not be considered in the future process. Algorithm 1 shows how to process and fuse data from cameras and sensors to realize OID. The computational complexity is $O(X \times Y)$, where $X$ and $Y$ are the numbers of detected objects and wearables, respectively. The detailed designs and implementations of data fusion algorithms in various scenario (without using IoTtalk) can be found in [4], [7], [37], [38].

IV. SYSTEM PROTOTYPE

We have developed a FusionTalk surveillance system based on the structure in Fig. 3. Several OID applications, such as location tracking, intrusion detection, and virtual fencing are demonstrated. In particular, user IDs can be visualized whenever possible. There are three cameras, twelve BLE beacons, two wearable devices, and three servers deployed in a meeting room. Their specifications are described as follows.

- Beacon: RF51822 ibeacon with transmission range of 30 meters and transmission interval of 100 ms.
- Wearable device: Raspberry Pi 3 Model B in a badge form equipped with an Accelerometer and Gyroscope sensor (MPU-6050), a compass (GY-271), and a Bluetooth module.
- Server: Intel Core i7-8700HQ processor with 32GB RAM and a GeForce GTX1080 Ti GPU.

We have developed three surveillance applications. Fig. 8(a) shows that we can keep tracking the locations of the employees with their badges, and easily visualize their movements and IDs, thus providing real-time awareness of various activities that are occurring within the surveillance region. Fig. 8(b) shows a scenario where an intruder (the one without a badge) is captured and tagged as “Unknown” on the screen, so an administrator can easily spot and track the intruder. Fig. 8(c)
Fig. 8. Application scenarios of FusionTalk: (a) Location tracking; (b) Intrusion detection; (c) Virtual fencing.

shows a virtual fence where an unauthorized person and a stolen bag are alerted. Thus real-time detection of unauthorized access to secure area and the theft of company assets can be monitored. The demo video is available in [39].

We make some remarks below. First, the above OID process is based on the users’ willingness to collaborate with our system. Nevertheless, our fusion system is still able to distinguish those people who collaborate and who do not. Second, if a device module does not collect enough data due to reasons such as occlusion and object detection errors, we will mark the corresponding object as “unknown”. We have implemented several data fusion algorithms in various scenarios, and conducted extensive experiments [4], [7], [37], [38], which show an identification accuracy above 95%. Demo videos are available in [40].

V. PERFORMANCE STUDY OF IoT DELAY

Since FusionTalk is a distributed system where the IoT devices interact using various wired and wireless channels, the delays of messages delivered from different input devices to a fusion output device significantly impact the execution of the OID applications. If the delays are too long, our system may associate a data source with another source in an asynchronous manner, thus deriving wrong OID results. This section conducts performance evaluation in various scenarios to investigate how communications delays affect the accuracy of identifying objects in FusionTalk, and presents the theoretical derivation and numerical analyses on the failure probability of FusionTalk under different network conditions.

To motivate our problem, let $t_1$ be the delay of a message $msg_1$ sent from the ObjTraj server (Fig. 3(1)) to DFusion server (Fig. 3(14)), i.e., the path (1) $\rightarrow$ (2) $\rightarrow$ (13) $\rightarrow$ (14) through Join 1, and let $t_2$ be the delay of a message $msg_2$ sent from ObjLoc (Fig. 3(5)) to DFusion (i.e., the path (5) $\rightarrow$ (6) $\rightarrow$ (13) $\rightarrow$ (14) through Join 3). DFusion associates $msg_1$ with $msg_2$ to derive the fusion results. The difference $|t_1 - t_2|$ should be small enough so that $msg_1$ and $msg_2$ are correctly paired. We conduct several experiments under the following scenarios.

- **Scenario 1**: The ObjTraj, the ObjLoc, and the DFusion server are placed in the same local area network (LAN) through wired connections.
- **Scenario 2**: The messages of ObjTraj and ObjLoc are sent to DFusion through 4G connections.

Fig. 9 shows the histograms of $t_1$ and $t_2$ in the two scenarios with 1000 measurements. We compute the expected values
A. Delay Analysis

We model the delay by an Erlang density function \( f_k(t_k) \), \( k = 1, 2 \), with the shape parameter \( n_k \) and the rate parameter \( \lambda_k \) \[41, 42\]:

\[
f_k(t_k) = \frac{\lambda_k^{n_k} t_k^{n_k-1} e^{-\lambda_k t_k}}{(n_k - 1)!},
\]

where \( E[t_k] = \frac{n_k}{\lambda_k} \) and \( V[t_k] = \frac{n_k}{\lambda_k^2} \). Without loss of generality, assume \( t_1 > t_2 \), so \( t_0 = (t_1 - t_2) > 0 \) and

\[
Pr[t_1 > t_2] = \int_{t_2=0}^{\infty} \int_{t_1=t_2}^{\infty} f_1(t_1)f_2(t_2)dt_1dt_2.
\]

The density function \( f_0(t_0|t_1 > t_2) \) can be expressed as

\[
f_0(t_0|t_1 > t_2) = \frac{f_0(t_0 \cap t_1 > t_2)}{Pr[t_1 > t_2]},
\]

where

\[
f_0(t_0 \cap t_1 > t_2) = \int_{t_2=0}^{\infty} f_1(t_2 + t_0)f_2(t_2)dt_2.
\]

Let

\[
\theta_i = \left( \begin{array}{c} n_2 + i - 1 \\ i \end{array} \right) \left( \frac{\lambda_1^i \lambda_2^{n_2}}{(\lambda_1 + \lambda_2)^{n_2+i}} \right).
\]

By Eq. (8) and (9), Eq. (7) can be expressed as

\[
f_0(t_0|t_1 > t_2) = \sum_{i=0}^{n_1-1} \left( \frac{\theta_i}{\sum_{i=0}^{n_1-1} \theta_i} \right) \left( \frac{\lambda_1^{n_1-i} t_0^{n_1-i-1} e^{-\lambda_1 t_0}}{(n_1 - i - 1)!} \right). \]

Let \( t_3 \) be the time period that the object is allowed to move within the tolerable localization error. We model it as a random variable with the density function \( f_3(t_3) \). If \( t_3 < t_0 \), the fusion results may be inaccurate. So

\[
Pr[t_3 < t_0|t_1 > t_2] = \sum_{i=0}^{n_1-1} \left( \frac{\theta_i}{\sum_{i=0}^{n_1-1} \theta_i} \right) \sum_{j=0}^{n_1-i-1} \left( \frac{\lambda_1^j \lambda_2^{n_1-i-j} e^{-\lambda_1 t_3}}{j!} \right) \int_{t_3=0}^{\infty} t_3^i f_3(t_3)e^{-\lambda_1 t_3}dt_3.
\]

Let \( f_3^*(s) \) be the Laplace transform for \( f_3(t_3) \). From the frequency-domain general derivative of Laplace transform, we have

\[
\int_{t=0}^{\infty} t^i f(t)e^{-st}dt = (-1)^j \frac{f^{(j)}(s)}{s^{j+1}}.
\]

Applying Eq. (12) to Eq. (11), we have

\[
Pr[t_3 < t_0|t_1 > t_2] = \sum_{i=0}^{n_1-1} \left( \frac{\theta_i}{\sum_{i=0}^{n_1-1} \theta_i} \right) \sum_{j=0}^{n_1-i-1} \left( \frac{(-\lambda_1)^j}{j!} \right) \left[ \frac{f_3^{(j)}(s)}{s^{j+1}} \right]_{s=\lambda_1}.
\]

Assuming that \( t_3 \) is exponentially distributed with the mean \( 1/\beta \) second, the Laplace transform \( f_3^*(s) \) can be expressed as

\[
f_3^*(s) = \frac{\beta}{s + \beta},
\]

and Eq. (13) is re-written as

\[
Pr[t_3 < t_0|t_1 > t_2] = \frac{1}{x} \left[ \frac{1}{\lambda_1 + \beta} \right] ^{n_1} \sum_{i=0}^{n_1-1} \theta_i \left( \frac{\lambda_1 + \beta}{\lambda_1} \right)^i.
\]

Substituting Eq. (9) into Eq. (15).

\[
Pr[t_3 < t_0|t_1 > t_2] = 1 - \frac{1}{\lambda_1 + \beta} \left[ \frac{n_2 + i - 1}{i} \right] x^i \text{ and } x = \frac{\lambda_1 + \beta}{\lambda_1 + \lambda_2}.
\]

Then

\[
(1 - x) y_{n_2} = 1 - \sum_{i=1}^{n_1-1} \left( \frac{n_2 + n_1 - 2}{n_1 - 1} \right) x^n + \sum_{i=1}^{n_1-1} \left( \frac{n_2 + i - 1}{i} \right) \left( \frac{n_2 + i - 2}{i - 1} \right) x^i.
\]
In Eq. (18), we have

\[
\binom{n_2 + i - 1}{i} - \binom{n_2 + i - 2}{i - 1} = \binom{n_2 + i - 2}{i}.
\]  
(19)

From Eq. (19), Eq. (18) is re-written as

\[(1 - x)y_{n_2} = y_{n_2 - 1} - \sum_{i=1}^{n_1-1} \binom{n_2 + n_1 - 2}{n_1 - 1} x^{n_1}.
\]  
(20)

From Eq. (20) we have

\[y_{n_2} = \left(\frac{1}{1 - x}\right) \left[y_{n_2 - 1} - \sum_{i=1}^{n_1-1} \binom{n_2 + n_1 - 2}{n_1 - 1} x^{n_1}\right],
\]  
(21)

where from Eq. (17) we have

\[y_0 = \sum_{i=0}^{n_1-1} x^i = \frac{1 - x^{n_1}}{1 - x}.
\]  
(22)

Therefore, Eq. (16) can be computed iteratively by using Eq. (21) and Eq. (22).

When \(n_1 = 1\) and \(n_2 = 1\), Eq. (16) is re-written as

\[Pr[t_3 < t_0] = \frac{\beta}{\lambda_1 + \beta}.
\]  
(23)

**B. Simulation Validation**

First, we validated Eq. (35) and (42) by simulations. We recalled the histograms of \(t_1\) and \(t_2\) under wired and 4G connections in Fig. 9. Based on the histograms, under wired connections, \(t_1\) and \(t_2\) can be approximated by Erlang distributions with \((n_1 = 36, \lambda_1 = 0.181)\) and \((n_2 = 21, \lambda_2 = 0.153)\), respectively. Under 4G connections, \(t_1\) and \(t_2\) can be approximated by Erlang distributions with \((n_1 = 20, \lambda_1 = 0.087)\) and \((n_2 = 15, \lambda_2 = 0.071)\), respectively. Then, we simulated the value of \(t_3\) for 1 million times and varied its mean from 0.1 to 1.5 seconds with a step factor of 0.01 seconds. Fig. 10 shows the discrepancies between the analytic results and the simulation results in the two scenarios. The results show that the discrepancies are within 1% for all cases when \(t_3 \geq 0.6\) seconds. Therefore, the analytic analysis is consistent with the simulations. Note that the simulation model follows the approach in [8], [28], [29], and the details are omitted.

Second, we measured the delays \(t_1\) and \(t_2\) and the failure probability \(Pr[t_3 < t_0]\) in wired and 4G scenarios. In the experiments, we varied \(t_3\) from 0.1 to 0.5 seconds with a step factor of 0.01 seconds. Fig. 11(a) shows the measurement results of \(Pr[t_3 < t_0]\) in the two scenarios. We observed that \(Pr[t_3 < t_0] < 0.01\%\) when \(t_3 \geq 0.45\) seconds. Therefore, the success probability of FusionTalk is sufficiently high (over 99.99%) to obtain the right fusion results when \(t_3\) is over 0.45 seconds in the two scenarios. These experiments indicate that the communication technologies used in the current FusionTalk version result in accurate pairing of object for fusion.

Third, we considered the worse case where communications are not stable by making the variance of delays larger than our current implementation. In the simulation, we set the expected values \(E[t_1] = 373.942\ ms\) and \(E[t_2] = 211.198\ ms\), varied the normalized delay variances \(V[t_1]/E[t_1]^2\) to 0.01, 0.1, or 1 \((k = 1, 2)\), and varied \(t_3\) from 0.1 to 1.5 seconds with a step factor of 0.01 seconds. Fig. 11(b) shows the failure probability \(Pr[t_3 < t_0]\) under these settings. The results show that \(Pr[t_3 < t_0]\) decreases with the decrease of the normalized delay variance, and that \(Pr[t_3 < t_0] < 1\%\) for all cases when \(t_3 \geq 1.4\) seconds. This indicates that FusionTalk can achieve a high success probability (over 99%) when \(t_3 > 1.4\) seconds even in communication systems with large variance.

**VI. CONCLUSION**

Previous OID systems were mostly developed under a static and centralized structure, which are difficult to adapt to distributed IoT devices and cameras with dynamic monitoring parameters. This paper describes a reconfigurable OID system named FusionTalk, which realizes better visualizability, flexibility, and extendibility by integrating the data fusion techniques with various distributed IoT devices based on IoTalk. With IoTalk, physical IoT devices and cyber IoT applications in FusionTalk are modularized and can be conveniently plugged in/out, reconfigured, and shared through GUIs.
Therefore, the scope of surveillance can be easily adjusted and an administrator can visualize not only the movement of a specific object in specific areas, but also the data of IoT device bundled with the object. The flexibility thus enables a lot of new surveillance applications. Through FusionTalk, our data fusion algorithm works well in various scenarios with an identification accuracy above 95%. In a distributed environment, FusionTalk is also able to adapt to various network delays with a failure probability less than 0.01%.

In the future, we would like to design various multi-sensor fusion algorithms and expand the FusionTalk system by accessing various IoT devices, such as Light Detection and Ranging (LiDAR) and robots. FusionTalk can play great roles in intelligent surveillance systems by quickly discovering, locating, and tracking target objects. As FusionTalk requires to fuse data from wearable devices, it is very suitable for armies, navigation tasks, health care teams, and highly controlled environments such as semiconductor factories.

**ACKNOWLEDGMENT**

This work has been supported by the Foundation item: National Key RD Program of China (2018YFB1402800); National Natural Science Foundation of China (61772560).

**APPENDIX A**

**DELAY ANALYSIS**

We model the delay by an Erlang density function $f_k(t_k)$, $k = 1, 2$.

$$f_k(t_k) = \frac{\lambda_n^k t_k^{n_k-1} e^{-\lambda_k t_k}}{(n_k - 1)!},$$ (24)

where $E[t_k] = \frac{n_k}{\lambda_k}$ and $V[t_k] = \frac{n_k}{\lambda_k^2}$. Without loss of generality, assume $t_1 > t_2$, so $t_0 = (t_1 - t_2) > 0$ and

$$Pr[t_1 > t_2] = \int_{t_2=0}^{t_1=\infty} f_1(t_1) f_2(t_2) dt_1 dt_2$$

$$= \int_{t_2=0}^{t_1=\infty} \left\{ \int_{t_1=0}^{t_2=\infty} \lambda_1^{n_1} t_1^{n_1-1} e^{-\lambda_1 t_1} dt_1 \right\} f_2(t_2) dt_2$$

$$= \sum_{i=0}^{n_1-1} \int_{t_2=0}^{t_1=\infty} \left( \frac{\lambda_1^i t_2^{n_2-1} e^{-\lambda_2 t_2}}{(n_2 - 1)!} \right) dt_2$$

$$= \sum_{i=0}^{n_1-1} \left( \frac{\lambda_1^i \lambda_2^{n_2} (n_2 + i - 1)!}{(n_1 - i)!} \right) \int_{t_2=0}^{t_1=\infty} \left( \frac{\lambda_2^{n_2} t_2^{n_2} e^{-\lambda_2 t_2}}{(n_2 - 1)!} \right) dt_2$$

$$= \sum_{i=0}^{n_1-1} \left( \frac{\lambda_1^i \lambda_2^{n_2} (n_2 + i - 1)!}{(n_1 - i)!} \right) \left( \frac{\lambda_1 \lambda_2^{n_2} t_2^{n_2 - 1} e^{-\lambda_2 t_2}}{(n_2 - 1)!} \right) dt_2$$

$$= \sum_{i=0}^{n_1-1} \left( \frac{\lambda_1^i \lambda_2^{n_2} (n_2 + i - 1)!}{(n_1 - i)!} \right) \left( \frac{\lambda_1 \lambda_2^{n_2} t_2^{n_2 - 1} e^{-\lambda_2 t_2}}{(n_2 - 1)!} \right) dt_2$$ (25)

The density function $f_0(t_0 | t_1 > t_2)$ can be expressed as

$$f_0(t_0 | t_1 > t_2) = \frac{f_0(t_0 \cap t_1 > t_2)}{Pr[t_1 > t_2]},$$ (26)

where

$$f_0(t_0 \cap t_1 > t_2) = \int_{t_2=0}^{t_1=\infty} f_1(t_2 + t_0) f_2(t_2) dt_2$$

$$= \int_{t_2=0}^{t_1=\infty} \left[ \frac{\lambda_1^{n_1} (t_2 + t_0)^{n_1-1} e^{-\lambda_1 (t_2 + t_0)}}{(n_1 - 1)!} \right] dt_2$$

$$= \frac{\lambda_1^{n_1} \lambda_2^{n_2} e^{-\lambda_2 t_2}}{(n_2 - 1)!} dt_2$$

$$= \frac{\lambda_1^{n_1} \lambda_2^{n_2} e^{-\lambda_2 t_0}}{(n_2 - 1)!} dt_2$$

$$= \sum_{i=0}^{n_1-1} \left( \frac{\lambda_1^{n_1} \lambda_2^{n_2} e^{-\lambda_2 t_0}}{(n_2 - 1)!} \right) \left( \frac{\lambda_1^{n_1} \lambda_2^{n_2} t_2^{n_2-1} e^{-\lambda_2 t_2}}{(n_2 - 1)!} \right) dt_2.$$ (27)

Let

$$\theta_i = \frac{n_2 + i - 1}{i} \left( \frac{\lambda_1 \lambda_2^{n_2} (n_2 + i - 1)!}{(n_1 - i)!} \right)$$

$$= \left( \frac{\lambda_2}{\lambda_1 + \lambda_2} \right)^{n_2} \left( \frac{n_2 + i - 1}{i} \right) \left( \frac{\lambda_1}{\lambda_1 + \lambda_2} \right)^i.$$ (28)

By Eq. (27) and (28), Eq. (26) can be expressed as

$$f_0(t_0 | t_1 > t_2) = \left( \frac{1}{Pr[t_1 > t_2]} \right) \sum_{i=0}^{n_1-1} \left( \frac{n_2 + i - 1}{i} \right)$$

$$\times \left( \frac{\lambda_1 \lambda_2^{n_2} (n_2 + i - 1)!}{(n_1 - i)!} \right) \left( \frac{\lambda_1 \lambda_2^{n_2} t_2^{n_2 - 1} e^{-\lambda_2 t_2}}{(n_2 - 1)!} \right) dt_2.$$ (29)

Let $t_3$ be the time period that the object is allowed to move within the tolerable localization error. We model it as a random variable with the density function $f_3(t_3)$. If $t_3 < t_0$, the fusion results may be inaccurate. So

$$Pr[t_3 < t_0 | t_1 > t_2] = \int_{t_3=0}^{t_3=\infty} f_3(t_3) f_0(t_0 | t_1 > t_2) dt_0 dt_3$$

$$= \sum_{i=0}^{n_1-1} \left( \frac{\theta_i}{\sum_{l=0}^{n_1-1} \theta_l} \right) \int_{t_3=0}^{t_3=\infty} f_3(t_3)$$

$$\times \left( \frac{\lambda_1 \lambda_2^{n_2} t_2^{n_2-1} e^{-\lambda_2 t_2}}{(n_2 - 1)!} \right) dt_2$$

$$= \sum_{i=0}^{n_1-1} \left( \frac{\theta_i}{\sum_{l=0}^{n_1-1} \theta_l} \right) \int_{t_3=0}^{t_3=\infty} f_3(t_3) \sum_{j=0}^{n_2-1} \left( \frac{\lambda_1^j t_2^j e^{-\lambda_2 t_2}}{j!} \right) dt_2.$$

$$= \sum_{i=0}^{n_1-1} \left( \frac{\theta_i}{\sum_{l=0}^{n_1-1} \theta_l} \right) \int_{t_3=0}^{t_3=\infty} f_3(t_3) \sum_{j=0}^{n_2-1} \left( \frac{\lambda_1^j t_2^j e^{-\lambda_2 t_2}}{j!} \right) dt_2.$$

(30)
Let $f_3^j(s)$ be the Laplace transform for $f_3(t_3)$. From the frequency-domain general derivative of Laplace transform, we have

$$
\int_0^\infty t^j f(t) e^{-st} \, dt = (-1)^j \left[ \frac{f^{(j)}(s)}{s^j} \right].
$$

(31)

Applying Eq. (31) to Eq. (30), we have

$$
Pr[t_3 < t_0|t_1 > t_2] = \frac{1}{\sum_{i=0}^{n_1-1} \left( \frac{1}{\beta} \right)} \sum_{i=0}^{n_1-1} \left[ \frac{\lambda_i^j}{(\lambda_1 + \beta)^{-1}} \right] \frac{f_3^j(s)}{s^j}.
$$

(32)

Assuming that $t_3$ is exponentially distributed with the mean $1/\beta$ second, the Laplace transform $f_3^j(s)$ can be expressed as

$$
f_3^j(s) = \frac{\beta}{s + \beta}.
$$

(33)

and Eq. (32) is re-written as

$$
Pr[t_3 < t_0|t_1 > t_2] = \sum_{i=0}^{n_1-1} \left( \frac{\theta_i}{\theta_i + \beta} \right) \sum_{j=0}^{n_1-1} \left[ \frac{\lambda_i^j}{(\lambda_1 + \beta)^j} \right] \frac{f_3^j(s)}{s^j}.
$$

(34)

Substituting Eq. (28) into Eq. (34),

$$
Pr[t_3 < t_0|t_1 > t_2] = 1 - \frac{1}{\sum_{i=0}^{n_1-1} \left( \frac{\lambda_1 + \beta}{\lambda_1 + \lambda_2} \right)} \sum_{i=0}^{n_1-1} \left( \frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2} \right) \frac{\lambda_2^{n_2}}{\lambda_1 + \lambda_2} y_{n_2},
$$

where

$$
y_{n_2} = \sum_{i=0}^{n_1-1} \left( \frac{n_2 + i - 1}{i} \right) x^i and x = \frac{\lambda_1 + \beta}{\lambda_1 + \lambda_2}.
$$

(36)

Then

$$
(1-x)y_{n_2} = 1 + \sum_{i=1}^{n_1-1} \left( \frac{n_2 + i - 1}{i} \right) x^i
$$

$$
= 1 + \sum_{i=1}^{n_1-1} \left( \frac{n_2 + 1 - 2}{i} \right) x^i
$$

$$
= 1 + \sum_{i=1}^{n_1-1} \left( \frac{n_2 + n_1 - 2}{n_1 - 1} \right) x^n + \sum_{i=1}^{n_1-1} \left( \frac{n_2 + i - 1}{i} \right) x^n.
$$

(37)

In Eq. (37), we have

$$
\frac{n_2 + i - 1}{i} - \frac{n_2 + i - 2}{i - 1} = \frac{(n_2 + i - 1) - (n_2 + i - 2)!}{(n_2 - 1)! (i - 1)!}
$$

$$
= \frac{(n_2 + i - 2)}{i - 1}
$$

$$
= \left( \frac{n_2 + 1 - 2}{i} \right) x^n.
$$

(38)

From Eq. (38), Eq. (37) is re-written as

$$
(1-x)y_{n_2} = \sum_{i=0}^{n_1-1} \left( \frac{n_2 + i - 2}{i} \right) x^i - \sum_{i=1}^{n_1-1} \left( \frac{n_2 + n_1 - 2}{n_1 - 1} \right) x^n
$$

$$
y_{n_2} = \sum_{i=1}^{n_1-1} \left( \frac{n_2 + n_1 - 2}{n_1 - 1} \right) x^n.
$$

(39)

From Eq. (39) we have

$$
y_{n_2} = \left( \frac{1}{1-x} \right) \frac{y_{n_2-1} - \sum_{i=1}^{n_1-1} \left( \frac{n_2 + n_1 - 2}{n_1 - 1} \right) x^n}{y_{n_2-1}},
$$

where from Eq. (36) we have

$$
y_0 = \sum_{i=0}^{n_1-1} x^i = \frac{1 - x^{n_1}}{1-x}.
$$

(41)

Therefore, Eq. (35) can be computed iteratively by using Eq. (40) and Eq. (41).

When $n_1 = 1$ and $n_2 = 1$, Eq. (35) is re-written as

$$
Pr[t_3 < t_0] = \frac{\beta}{\lambda_1 + \beta}.
$$

(42)

REFERENCES


Kun-Ru Wu received his B.S. degree in Electrical Engineering from the National Chung Hsing University, Taiwan. He obtained his Ph.D. degree from the Institute of Communications Engineering at National Chiao Tung University (NCTU) in 2015. Dr. Wu is currently an assistant researcher with the Department of Computer Science, NCTU, Taiwan. His research interests include system programming, sensor fusion, wireless mesh networks and linux-based system prototyping.

Yi-Bing Lin (M’96-SM’96-F’03) is Winbond Chair Professor of National Chiao Tung University (NCTU). He received his Bachelor’s degree from National Cheng Kung University, Taiwan, in 1983, and his Ph.D. from the University of Washington, USA, in 1990. From 1990 to 1995 he was a Research Scientist with Bellcore (Telcordia). He then joined NCTU in Taiwan, where he remains. In 2010, Lin became a lifetime Chair Professor of NCTU, and in 2011, the Vice President of NCTU. During 2014-2016, Lin was Deputy Minister, Ministry of Science and Technology, Taiwan. Since 2016, Lin has been appointed as Vice Chancellor, University System of Taiwan (for NCTU, NTHU, NCU, and NYM). Lin is AAAS Fellow, ACM Fellow, IEEE Fellow, and IET Fellow.

Lingyan Zhang is an assistant professor of Central South University, China. She received her B.S. degree in Network Engineering from Beijing University of Posts and Telecommunications (BUPT) in 2013, and received her Ph.D degree in Computer Science and Technology from BUPT in 2018. From 2018 to 2020, she was a postdoctoral researcher in the Department of Computer Science, National Chiao Tung University, Taiwan. Her research interests include IoT, mobile computing, cloud computing, service computing, quality of experience.

Hung-Cheng Lin received the B.S. in Computer Science and Engineering from National Sun Yat-sen University, Taiwan, in 2018. He is currently working towards the M.S. in computer science at National Chiao Tung University. His current research interests include machine learning, mobile computing, and Internet of Things.

Yu-Chee Tseng received the Ph.D. degree in computer and information science from The Ohio State University in 1994. He was the Chairman with the College of Computer Science, National Chiao-Tung University, Taiwan, from 2005 to 2009, where he has been the Dean since 2011. He has been the NCTU Chair Professor since 2011. He was the Y.Z. Hsu Scientific Chair Professor from 2012 to 2013. His research interests include mobile computing, wireless communication, and Internet of Things. He received the Outstanding Research Award (National Science Council, 2001, 2003, and 2009), the Best Paper Award (Intel. Conf. on Parallel Processing, 2003), the Elite I. T. Award in 2004, the Distinguished Alumnus Award (Ohio State University, 2005), and the Y. Z. Hsu Scientific Paper Award in 2009. He served/serves on the Editorial Boards of the IEEE Transactions on Vehicular Technology, the IEEE Transactions on Mobile Computing, the IEEE Transactions on Parallel and Distributed Systems, and the IEEE Internet of Things Journal. His h-index is over 50.