Sensors and Materials, Vol. 34, No. 8 (2022) 3063–3091
MYU Tokyo

S & M 3023

ParmoSense: Scenario-based Participatory Mobile Urban Sensing Platform with User Motivation Engine

Yuki Matsuda,[1*] Shogo Kawanaka,[1] Hirohiko Suwa,[1] Yutaka Arakawa,[2] and Keiichi Yasumoto[1]

[1]Graduate School of Science and Technology, Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma city, Nara 630-0101, Japan
[2]Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan

(Received April 30, 2022; accepted June 16, 2022)

Keywords: civic computing, ubiquitous computing, mobile computing, participatory sensing, smart city, urban sensing, gamification, incentive mechanism

The rapid proliferation of mobile devices with various sensors has enabled participatory mobile sensing (PMS). Several PMS platforms suffer from open issues including the limited use of their functions to a specific scenario/case and the necessity of technical knowledge for organizers. This paper proposes a novel PMS platform named ParmoSense for easy and flexible data collection. To reduce the burden on both organizers and participants, we employ two novel features. First, essential PMS functions implemented as modules can be easily chosen and combined for sensing in different scenarios. Second, the scenario-based description feature allows organizers to easily and quickly prepare a new instance of PMS and enable people to easily participate in the sensing. It also provides multiple functions to motivate participants for sustainable operation. Through a performance comparison with existing PMS platforms, we confirmed that ParmoSense shows the best cost performance in terms of the workload for preparation and the variety of functions. In addition, to evaluate the availability and usability of ParmoSense, we conducted 19 case studies over four years with ordinary citizens. As the result of a questionnaire survey carried out during the case studies, we confirmed that ParmoSense can be easily operated by ordinary citizens without technical skills.

1. Introduction

Mobile devices are equipped with various sensors including a GPS, inertial sensors, environment sensors, camera, and microphone. The rapid spread of such mobile devices has enabled participatory mobile sensing (PMS).[(1–3)] PMS systems are based on crowdsourcing technology, where data in a wide geographical area can be collected efficiently and at low cost by leveraging sensors on mobile devices carried by ordinary citizens. Many applications can utilize collected urban data to bring various benefits to our daily lives. For instance, because PMS systems use common devices, they can be easily used to collect data for urban analysis,[(4)]

*Corresponding author: e-mail: yukimat@is.naist.jp
https://doi.org/10.18494/SAM3961

office management,(5) healthcare,(6) and education.(7,8) In addition, since PMS systems can be applied to any region in which people stay or pass through, they are very effective for collecting geospatial data over a wide area. In an urban environment, for example, data such as illuminance of the road at night,(9) noise levels in the city,(10,11) and air pollution levels(12,13) can be collected.

PMS is a sensing mechanism based on the voluntarism of general people. In other words, the sustainability of the system is a critical challenge in real-world operations. As an idea to enhance this sustainability, mutual linkage with the local community in which ecosystems are already formed can be considered. For example, civic cooperation based on the idea that people work with government, universities, companies, and so forth to promote community development is spreading globally. In particular, in recent years, civic technology (CivicTech), which combines ICT and civic cooperation, is gathering attention, such as FixMyStreet.(14) In our study, we focus on PMS systems that can be used in CivicTech communities.

To realize PMS systems in the real world for broad urban environment analysis, we believe that a platform is needed that can be easily and quickly customized by organizers to perform a variety of sensing tasks and is easy to set up and run on participants' smartphones. However, when we investigated the functions implemented on existing PMS platforms,(15–23) we found two main challenges in using these platforms for broad urban environment analysis: challenge C1: limited support of essential functions and challenge C2: difficulty of system construction and operation.

Regarding C1, existing platforms tend to focus on specific sensing purposes, for example, urban transport data sensing, and therefore support limited functions. Because the purpose of sensing differs among organizers of urban sensing, the necessary functions, i.e., sensing function, incentive mechanism, task request control, and data processing method, will also depend on the purpose. Thus, for an ideal PMS platform, flexibility to adapt the platform to perform sensing for various purposes is mandatory. Moreover, motivating participants is an important aspect of PMS since participatory sensing relies on voluntary participation of ordinary citizens.(24) However, we found that these platforms do not sufficiently motivate them. Regarding C2, the platforms require a high level of technical skill for users. For example, some platforms require programming skills for organizers and data processing skills for participants. To open the door of participatory sensing to non-technical users, it is necessary to ensure that PMS systems can be easily constructed and operated by both organizers and participants.

In this study, we designed and built a novel PMS platform named ParmoSense for easily and flexibly collecting urban environmental information for various purposes by overcoming the challenges mentioned above. To achieve this, we employ two features: modularization of functions and scenario-based PMS system description. We provide various functions essential for PMS systems such as sensing functions, motivating functions for participants, and processing functions for collected data, and allow organizers to combine these modularized functions freely through a graphical user interface (GUI) web application. We call a combination of these modularized functions a scenario. Once a scenario has been created, participants can download it onto the ParmoSense client application and run it without doing any further setup or processing tasks. Thus, participants can contribute to many different sensing tasks without installing multiple applications or performing complicated tasks that require technical skills.

To evaluate the superiority of ParmoSense, we compared its performance with existing PMS platforms. First, we confirmed that ParmoSense provides a greater variety of functions than the existing PMS platforms, which solves C1. We also found that ParmoSense makes it easier for organizers to prepare PMS systems. It belongs to the group with the low preparation workload among the existing platforms, which solves C2. From both perspectives, therefore, ParmoSense shows the best cost performance. In addition, to evaluate the availability and usability of ParmoSense in the real world, we conducted 19 practical case studies with ordinary citizens including non-technical people. We confirmed that ParmoSense can deal with various sensing targets, organizers, and participants in real environments.

Our contributions in this paper are as follows:

i. We designed a PMS platform, named ParmoSense, which allows ordinary people to easily operate PMS systems with scenario-based system construction regardless of their technical skills.
ii. We organized functionality requirements through a survey of existing PMS platforms and implemented all functions as combinable modules.
iii. We confirmed that ParmoSense shows the best cost performance in terms of the variety of functions and preparation workload through a comparison with nine existing platforms.
iv. We confirmed the availability and usability of ParmoSense through 19 practical case studies in the real world and interviews with participants and organizers of sensing tasks.

The rest of this paper is organized as follows. In Sect. 2, we survey the existing PMS platforms and systems and organize the functions required for PMS systems and the skill requirements for participants and organizers. In Sect. 3, we describe the concept and architecture of ParmoSense and the functions provided on ParmoSense. To evaluate ParmoSense, we compare its functionality and performance with existing PMS platforms in Sect. 4, and we conduct 19 practical case studies with general people including non-technical persons in Sect. 5. Section 6 concludes this paper, in which we discuss the limitations and future challenges of ParmoSense.

2. Related Work and Challenges

This section is devoted to clarifying the types of functions necessary for PMS systems and the types of skills required for users (organizers and participants) of PMS systems. We first organize the necessary functions into three categories: sensing functions, motivating functions, and processing functions. Then, we investigate the functions implemented in existing PMS platforms.(15–22) The results are summarized in Table 1. The skills required by organizers and participants for existing PMS platforms are shown in Table 2.

2.1 Functions essential in PMS platform

2.1.1 Sensing functions

Sensing is an essential part of PMS systems. We define sensing functions as those that allow the organizer to specify the types of sensors to use and how to collect sensing data from the

Table 1
Overview of supported functions.

Platforms	Functions							
	Sensing functions		Motivating functions			Processing functions		
	F1 Implicit sensing	F2 Explicit sensing	F3 Request	F4 Reward	F5 Feedback	F6 Editing	F7 Browsing	F8 Export
AWARE (15)	✓	△*b	△*e		△*g		✓	✓
Sensus (16)	✓	△*c					✓	△*h
Medusa (17)	✓	△*d	△*e	△*f			✓	
Funf (18)	✓	△*d				✓	✓	✓
MinaQn (19)	△*a	△*b	△*e		✓		✓	
KOKOPIN app (20)	△*a	△*c			✓		✓	✓
Ohmage (21)	△*c	✓			△*g	✓	✓	✓
OpenDataKit (22)	△*a	✓				✓	✓	✓
GP-Selector (23)	✓	✓	✓	△*f				
ParmoSense	✓	✓	✓	✓	✓	✓	✓	✓

*a Location (GPS) only supported. *b Media upload (photo, sound, etc.) limited. *c Raw data collection unsupported. *d Questionnaire unsupported.
*e Static request only supported. *f Monetary incentives only supported. *g Feedback of data collected by oneself only supported. *h command-line tool is required.

Table 2
Overview of platform skill requirements.

Platforms	Skill requirements			
	Organizer skills		Participant skills	
	R1 Development	R2 App distribution	R3 App/Func. management	R4 Data processing
AWARE (15)	As needed*a	-	Required	-
Sensus (16)	As needed*b	-	-	-
Medusa (17)	Required	Required	-	-
Funf (18)	-	-	Required	Required
MinaQn*c (19)	-	-	-	-
KOKOPIN app*c (20)	-	-	-	-
Ohmage (21)	-	-	Required	-
OpenDataKit (22)	As needed*a	Required	Required	Required
GP-Selector (23)	As needed*d	-	-	-
ParmoSense	-	-	-	-

*a Development by programming is needed for extending functions. *b Database server is required for data collection.
*c These platforms assume the use of ordinary citizens or administrative officers, hence, technical skills are not required.
*d XML-based script coding is needed for making complex participant selection constraints.

urban environment. There are two different ways of sensing: implicit sensing (F1) and explicit sensing (F2).

Implicit sensing (F1): Uses sensors embedded in mobile devices. It is mainly used for collecting urban environmental data without actively involving the participant, i.e., implicit sensing.

Explicit sensing (F2): Used for collecting data generated by human behavior, e.g., photos, voices, and questionnaires. It is used for collecting urban environmental data through directly involving participants, i.e., explicitly, and locally.

AWARE(15) provides a platform for both implicit and explicit sensing. For implicit sensing, the organizer can choose which smartphone sensors to use from a web UI. They can also configure the detailed settings such as the sensing interval. For explicit sensing, AWARE allows the organizer to distribute questionnaires manually.

Ohmage(21) supports explicit sensing by allowing participants to post reports by themselves. Several report formats are accepted such as single/multiple selections, free text, and multimedia (e.g., photo, sound). Ohmage also supplements collected data through implicit sensing. Specifically, it can be used to record the transportation status (e.g., still, walking, running) of a participant.

However, the other conventional platforms tend to focus more on either implicit or explicit sensing (as shown in Table 1), and provide limited functionality for other forms of sensing. As mentioned before, the two sensing methods have many differences such as the data type that can be collected and the data's features. Thus, with these platforms, it is difficult to supplement the collected data due to severe restrictions in sensing functions.

In this paper, to realize a flexible sensing platform, we aim to provide functions for both sensing methods comprehensively and support the organizers' ability to easily choose and combine functions.

2.1.2 Motivating functions

Since PMS relies on the voluntary participation of ordinary citizens, it is essential to focus on not only acquiring users, but also motivating them to continue participating in the sensing tasks, i.e., to support user retention and activation.(25) Motivating functions allow the organizer to define the methods for motivating participants. Some of the conventional platforms use outside stimuli to support participant motivation and engagement. We classify these outside stimuli as follows:

Requests (F3): This approach encourages behavior by explicitly requesting participation. There are many methods of sending requests. The most common method is static/dynamic requests, which include providing a task list, issuing notifications, and so forth. Other methods such as audition(17) and reverse auction,(26,27) which purposely restrict the rights of contribution and make participants scramble to contribute, have also been used. In addition, willingness-based participant selection,(23) in which the participants to be requested are selected on the basis of their estimated willingness to carry out the sensing task, has been used.

Rewards (F4): In this method, participants are compensated for their contribution through monetary and non-monetary incentives.(28,29) A monetary incentive encourages participants to contribute to the system by giving, for example, in-app currency/points, discount coupons, and gifts. Participants can directly obtain explicit value in return for their contribution. This approach often has high effectiveness; however, there are problems related to the sustainability of system management.(24) To reduce the cost of rewards, the auction mechanism(30) and a non-monetary incentive mechanism can be used. A non-monetary incentive gives an enjoyable experience as compensation for the participant's contribution.(31–33) Types of experiences include interaction with other participants and gamification. Interaction with other participants induces social facilitation effects that stimulate the participant to contribute more actively(34) to obtain praise from others, such as more "likes" and comments. Gamification is a mechanism that introduces game elements into a conventional system, and it has been shown to contribute to the motivation of participants and enable the reduction of monetary rewards.(35)

Feedback (F5): This method encourages participation by providing feedback such as a visualization of the participant's contribution on a map, graph, or timeline. Sometimes the contributions of other participants are also included in the visualizations.

MinaQn[19] uses a recruitment mechanism to grant participants the right to participate in urban planning (contribution to society) as a non-monetary incentive. In addition, the platform also visually summarizes the participant's contribution to increase their willingness to continue contributing.

Medusa[17] is a platform that utilizes monetary incentives effectively. Medusa can acquire participants using recruitment, which provides money as compensation, and through audition. Furthermore, to retain participants, Medusa adopts the concept of reverse incentive (obligation/responsibility of executing tasks), where workers pay organizers for the privilege of performing the task. This can help prevent participants from quitting the system in the middle of sensing tasks.

GP-Selector[23] is a platform that employs a participant selection algorithm by considering various conditions suitable for sensing tasks. It includes location-based constraints (e.g., geofences), capability-based constraints (e.g., available sensors), and willingness prediction.

In these conventional platforms, the functions to motivate participants have a number of shortcomings. For example, with request functions (F3), to increase the number of successful requests, it is necessary to consider the notification timing and the target participants, but this is not supported. Similarly, with reward functions (F4), we need to consider not only monetary incentives, but also gamification. In this paper, we consider how to design motivating functions that incorporate the concepts of interruption through notification and gamification.

2.1.3 Processing functions

In general, organizers intend to analyze or visualize urban environmental data collected with PMS. Hence, PMS platforms must support easy and quick access to this data. In the processing functions, the organizer defines methods of data processing to be used. The following functions are implemented in conventional platforms:

Data editing (F6): This involves data cleansing and labeling as pre-processing for detailed data analysis.

Data browsing (F7): This involves monitoring the status of data collection and visualizing the collected data.

Data export (F8): This involves exporting the collected data for more detailed analysis and/or visualization with third-party tools. Various exporting format types are supported depending on the purpose, such as CSV, JSON, XML, and RDB.

Funf[18,36] and OpenDataKit[22] are platforms that mainly focus on data cleansing and visualizing, as well as exporting. Additionally, these platforms support data processing in a variety of environments such as in the cloud and on the smartphones used for data collection (endpoint devices). Owing to the various processing functions on the platforms, they have been utilized in many research projects. Although most of these platforms also support basic functions, there are differences from the functions supported by other conventional platforms.[15,19,21]

In this paper, we implement all functions (F6–F8) as in Funf[18,36] and OpenDataKit.[22] During implementation, we consider how to reduce the technical skills required for organizers and participants. Specific skills needed to operate existing platforms are described in the next section.

2.2 Skills required for operation and use

In PMS, it is assumed that the organizer may be from a non-technical profession/background, e.g., they may be an administrative officer or urban planner, and that ordinary citizens participate in the data collection. Thus, it is necessary to reconsider the skills required by the PMS platforms for the organizers and participants.

Table 2 shows the skill requirements for each conventional platform. Development skills (R1) and App distribution skills (R2) are required for organizers:

Development skills (R1): Skills to develop the urban sensing system for a specific purpose.

App distribution skills (R2): Skills to distribute client applications to participants' smartphones.

AWARE,(15) Medusa,(17) and OpenDataKit(22) have high extendibility as a framework, but a high level of programming skill is required for system construction (R1). In addition, most of the systems that require development also require organizers to have the skills necessary to release applications on official stores such as Google Play and AppStore (R2). With web-based platforms such as MinaQn,(19) on the other hand, deploying the applications is straightforward. However, it is necessary to continuously attract users to the web application (R2).

App/function management skills (R3) and data processing skills (R4) are other skills required from participants:

App/function management skills (R3): Managing applications and functions such as the installation of applications and setting of functions to accomplish sensing tasks.

Data processing skills (R4): Processing data collected using the participant's device before uploading.

Since AWARE,(15) Ohmage,(21) and OpenDataKit(22) are platforms with many functions, the functions are divided into multiple applications and provided to participants. Also, Funf(18) requires the participants themselves to set up the sensors (e.g., the sensing interval) for each smartphone. Therefore, to construct the sensing environment that the organizer intended, knowledge and skills related to applications and functions are required for participants (R3).

Funf(18) and OpenDataKit(22) adopt a mechanism where they ask participants to perform data processing and cleansing. Such processing requires knowledge and skill to judge the usefulness of the collected data, so the burden on participants is substantial (R4).

Overall, conventional platforms require various skills for both organizers and participants to construct and operate the system. To realize PMS systems that can be used by non-technical people, these problems must be addressed. In this study, we propose a new platform to resolve these problems.

3. Scenario-based Participatory Mobile Sensing Platform

We design and implement a novel participatory urban sensing platform, ParmoSense, to solve the problems mentioned in Sect. 2. ParmoSense aims to solve the following key challenges:

C1: Limited support of essential functions needed for PMS systems (listed in Table 1)

C2: Difficulty of system construction and operation

ParmoSense allows organizers and participants to operate or contribute to PMS systems without complex procedures or technical skills.

3.1 Basic principles of ParmoSense

ParmoSense is based on the following three basic principles.

Principle #1 Modularized functions: ParmoSense must achieve two contradictory requirements: easiness of system construction and diversity of available functions. Therefore, we employ the idea of modularizing functions inspired by existing research.(15,21) ParmoSense provides the sensing, motivating, and processing functions in Table 1, and these can be combined to form a PMS system.

Principle #2 Standardized PMS system: Conventional platforms require a high level of technical skills, such as programming skills for organizers to customize the platform for a specific sensing task. To deal with multiple purposes flexibly, ParmoSense is composed as a combination of modularized functions as well as the detailed settings of each function. We unify the combination and settings as a scenario. A scenario contains information such as the scenario name, description, sensing targets, sensing area, period, and motivation method. Through the GUI tool in ParmoSense, the organizer can generate the scenario easily. On the basis of the created scenario, ParmoSense automatically configures both a server system and a client application by distributing necessary information. Since scenarios can be created using any combination of functions and settings, logically, any type of PMS system can be built with ParmoSense. Another important advantage of the scenario-based system is that users can participate in various PMS projects through one client application, whereas in the past, each PMS project required a dedicated application.

Principle #3 Customizable motivation engine: The most effective way of motivating participants depends on the purpose of sensing. To realize sustainable urban environmental sensing, ParmoSense has a motivation engine with a variety of motivation algorithms. The motivation engine provides the following functions for motivating participants:

- Motivation based on the behavior of an individual: Granting incentives according to the contribution and visualization of the contribution.
- Motivation for all participants, regardless of contribution: Providing competition mechanisms such as rankings and sharing of experiences among participants.

Additionally, by considering temporal/spatial information, such as the current time and position of the participant, it is possible to control the actuation timings of these functions. The optimal motivating algorithm according to the purpose of the organizer can be incorporated into the PMS system by combining and customizing these functions.

3.2 Overview of ParmoSense system

The architectural design of ParmoSense is shown in Fig. 1. ParmoSense consists of three parts that have the following roles:
ParmoSense dashboard: A web application for organizers that can be used to create and distribute scenarios of PMS systems and to process collected data.
ParmoSense client: A client application for participants that can run various scenarios. By downloading and installing a scenario, it behaves as the corresponding sensing application.
ParmoSense server: A central system for integrated management of the scenarios created by organizers and for automatically constructing a virtual server system for each scenario. It can collect sensing data from the ParmoSense client and generate feedback based on the analysis of the collected data.

Arrows in Fig. 1 show contents transferred between the organizer/participant and the ParmoSense server for each operation phase of PMS. Each phase is defined as follows:

i. **Distributing phase:** The phase for distributing the scenario created by the organizer to the participants via the ParmoSense server.
ii. **Sensing phase:** The phase for giving feedback (e.g., a reward) to the contribution of the participant such as uploading sensing data by participants.
iii. **Processing phase:** The phase for editing, cleansing, and visualizing the collected data.

Since ParmoSense is based on Principles #1 and #2 mentioned above, organizers can distribute sensing applications by simply exchanging scenarios with participants in the

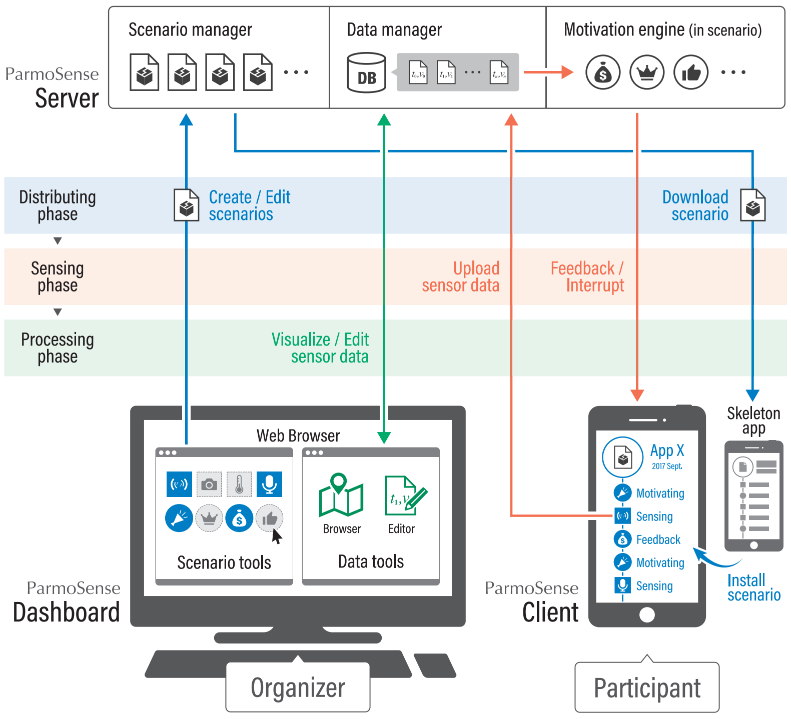


Fig. 1. (Color online) Design concept of ParmoSense.

distributing phase. It is therefore unnecessary for each participant to manage the applications by themselves. Furthermore, owing to Principle #3, in the sensing phase, the feedback for motivating participants is created by the motivation engine using the collected data and automatically provided to participants.

3.3 ParmoSense system architecture

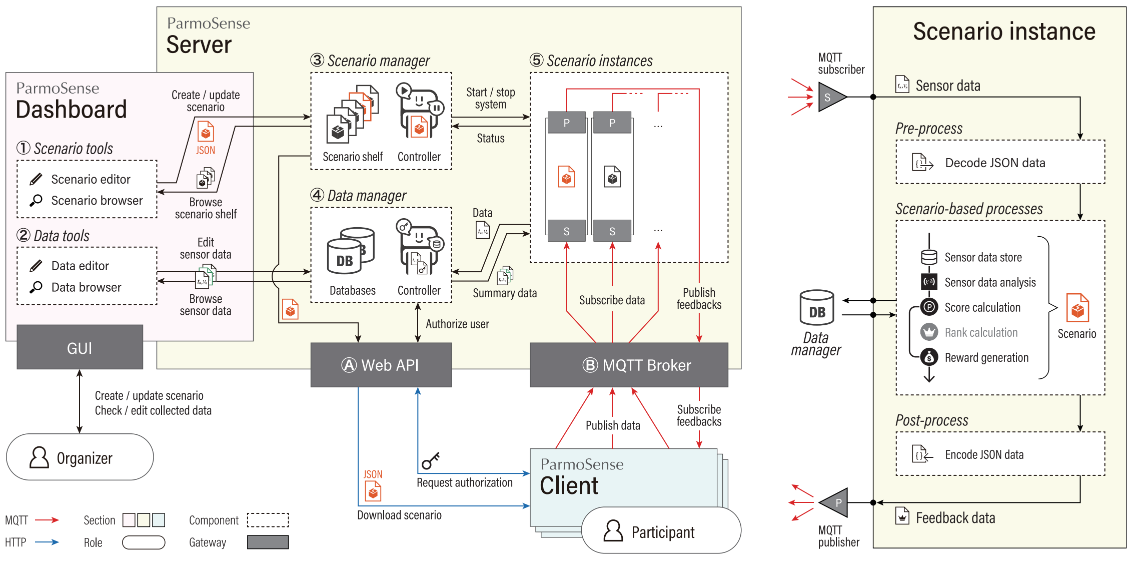
The concrete system configuration of ParmoSense is shown in Fig. 2. In the following subsections, we describe the ParmoSense dashboard used by the organizer, the ParmoSense client used by participants, and the ParmoSense server in more detail.

3.3.1 ParmoSense dashboard

An organizer carries out every operation, such as management of a PMS scenario and processing of collected data on a web application named the ParmoSense dashboard. The dashboard consists of Scenario Tools and Data Tools shown in Fig. 2(a) (1) and (2), respectively.

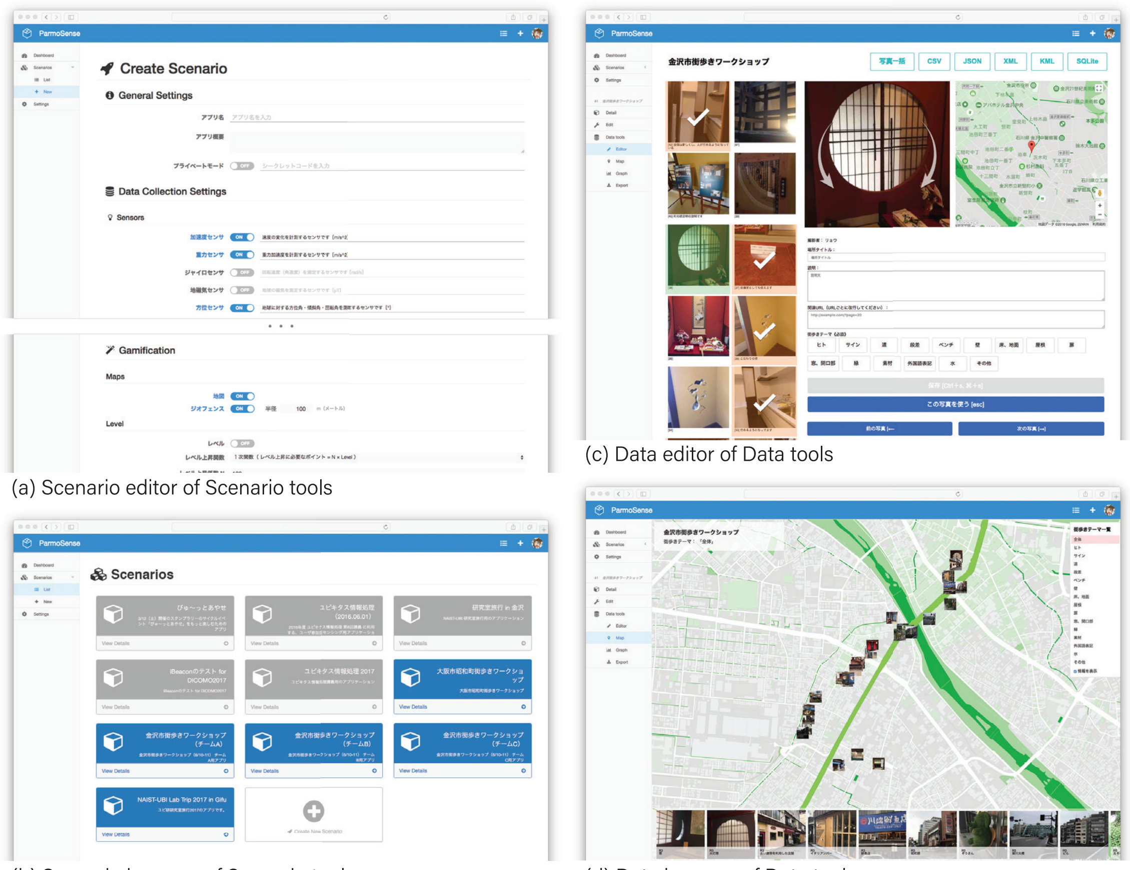
Scenario Tools [Fig. 2(a) (1)] provide many operations such as creating, editing, and deleting PMS scenarios and browsing, activating, and stopping scenarios. Figure 3(a) shows the user interface for editing scenarios. The organizer can describe a scenario using the three types of functions (sensing, motivating, and processing functions) mentioned in Sect. 2 without programming through the GUI (Principles #1, #3). The scenario defined in Scenario Tools is automatically converted to JSON format and transferred between each part of ParmoSense.

When the scenario editing is completed, the virtual server system is automatically built depending on the scenario and deployed by Scenario Manager [Fig. 2(a) (3)]. At the same time,



(a) Whole system architecture (b) Scenario instance architecture

Fig. 2. (Color online) System architecture of ParmoSense. (a) ParmoSense consists of three components: a dashboard, server, and client; (b) scenario instances are built for each PMS scenario.



(a) Scenario editor of Scenario tools

(b) Scenario browser of Scenario tools

(c) Data editor of Data tools

(d) Data browser of Data tools

Fig. 3. (Color online) User interface of ParmoSense dashboard. (a and b) Scenario Tools interface as shown in Fig. 2. (a) (1); (c and d) Data Tools interface as shown in Fig. 2. (a) (2). (The screenshot is extracted from Google Map.)

the QR code for downloading the scenario to the ParmoSense client is automatically generated. Figure 3(b) shows the user interface for browsing/managing the scenario created by organizers. Scenarios currently in progress/stopped are indicated in blue/gray, respectively, and these statuses and the scenario settings can be changed by using the GUI.

Data Tools [Fig. 2(a) (2)] provide the functions for processing and visualizing data aggregated into Data Manager [Fig. 2(a) (4)]. The user interface for editing the collected data is shown in Fig. 3(c). The organizer can improve the quality of the data by editing/excluding inappropriate data from the collected data. The organizer can also add labels to the data. The processed data can be exported in the form of JSON, CSV, RDB, and so forth. The user interface for visualizing the collected data is shown in Fig. 3(d). The organizer can check the data in two ways: overlaying them on a geographical map or sorting them in a time-series order.

3.3.2 ParmoSense client

The participant performs all sensing tasks of ParmoSense through the ParmoSense client smartphone application. The ParmoSense client runs on smartphones with Android OS or iOS, and it can be installed from general application stores (Google Play, AppStore). Since the behavior of the PMS system on ParmoSense is defined by a scenario (Principle #2), it can behave as various sensing applications by installing scenarios to the ParmoSense client.

Figures 4(a) and 4(b) show the user interface for the scenario installation. The participant performs the following steps to install the application:

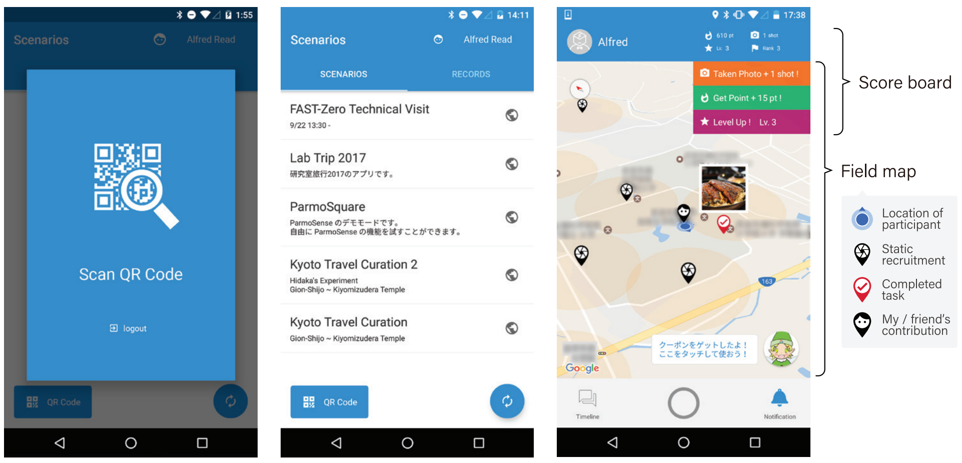
i. Log in to the ParmoSense client via Google Authentication.
ii. Scan the scenario QR code by using the participant's smartphone camera [Fig. 4(a)]. An organizer can obtain QR codes of scenarios from the ParmoSense dashboard and print them out for participants.
iii. The participant confirms their participation in sensing.

This procedure makes it easy to install scenarios, and the entire procedure is done via the Web API shown in Fig. 2(a) (A). Participants can participate in multiple scenarios. The scenarios that have been installed and performed in the past are listed, as shown in Fig. 4(b). This makes it easy for a participant to join the same scenario again.

Figure 4(c) shows an example of the ParmoSense client interface after installing the scenario and participating in sensing tasks. Requests of static sensing tasks are shown as pins on the map, and the participants can carry out the task at each location and acquire the reward accordingly. The user score is displayed in the upper right area, which is a means of providing feedback to the participant through gamification and visualization of contributions. The feedback and the execution status of tasks are reflected in real time according to the participant's and other participants' actions via the MQTT (MQ Telemetry Transport)(37) broker shown in Fig. 2(a) (B). ParmoSense realizes many-to-many and real-time communication by adopting MQTT as a communication protocol.

3.3.3 ParmoSense server

The ParmoSense server consists of three parts: Scenario Manager, Data Manager, and Scenario Instances, shown in Fig. 2(a) (3), (4), and (5), respectively.



(a) Scenario installation (b) Installed scenario list (c) Main PMS view (map view)

Fig. 4. (Color online) User interface of ParmoSense client. (a) Participants can install a PMS scenario via scanning the QR code; (b) installed scenarios are listed for switching scenarios; (c) PMS tasks and participants' contributions are shown on a map view. (The screenshot is extracted from Google Map.)

The scenario manager stores the scenario created on the ParmoSense dashboard and constructs and manages each scenario instance in accordance with the scenario. A scenario instance is an instance executed on a server that communicates with a ParmoSense client. By automatically constructing this scenario instance for each scenario and forming scenario instances, various scenarios in ParmoSense can be created without programming. The scenario manager monitors the operation status of the scenario instance, and the organizer can start or stop the operation. It also detects unexpected problems in the scenario instance, and it stops or restarts the instance. The data collected by participants is aggregated in Data Manager, and this data is used for calculating the participants' score, visualization on the map, and so forth.

Figure 2(b) shows the mechanism of the scenario instance. The scenario manager builds the scenario instance by incorporating the module program of corresponding functions (sensing functions, motivating functions, processing functions) based on the scenario that an organizer has created. The scenario instance communicates with the ParmoSense client via MQTT as described above. If a participant sends (publishes) the sensing data, the corresponding scenario instance receives (subscribes to) the data collected by the participant's smartphone sensors and processes the data using modularized functions (e.g., analysis of data, calculation of ranking) described in the scenario. According to these results, response data is generated and published to all participants who should be informed.

3.4 Functions supported

ParmoSense comprehensively supports the functions listed in Table 1. In this section, we outline the support status of each function.

Implicit sensing (F1): ParmoSense supports data collection from sensors embedded in smartphones. The types of supported sensors are:

- Position sensors (e.g., GPS)
- Environmental sensors (e.g., light sensor, barometer)
- Inertial sensors (e.g., accelerometer, gyroscope)
- External sensor devices (e.g., heart rate sensor)

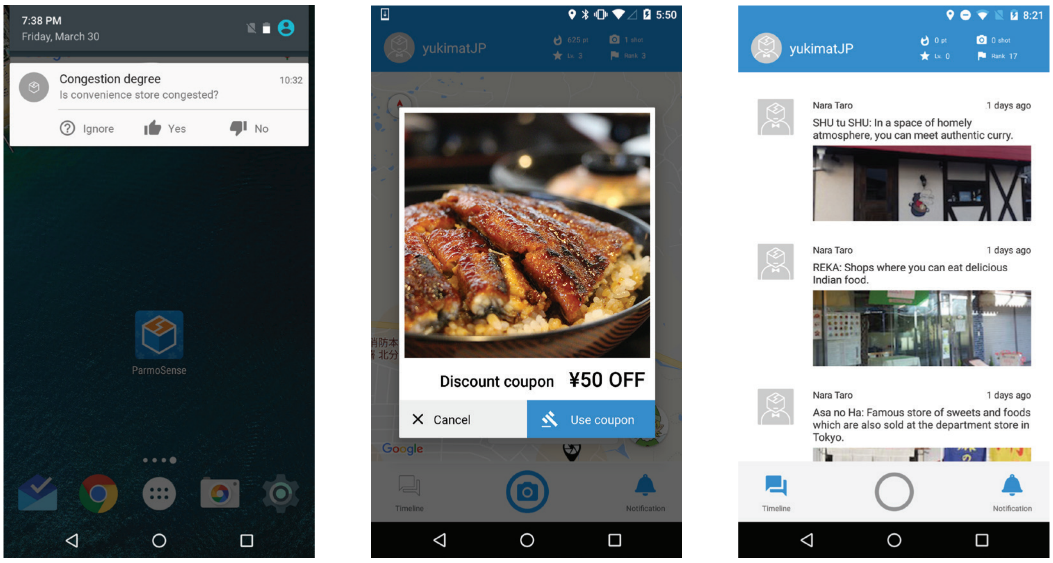
It also supports data collection of Bluetooth Low Energy (BLE) scan logs of peripheral devices (e.g., iBeacon, other smartphones). For all these sensors, detailed configurations such as the measurement interval and enabling/disabling of background measurement can be set on the scenario editor of the ParmoSense dashboard by the organizer.

Explicit sensing (F2): ParmoSense supports various types of methods of collecting data that cannot be collected by sensors embedded in a smartphone. One of them is photo uploading. When taking and uploading photos, participants can provide additional data such as explanatory texts of the photo taken, GPS position, and other data obtained by implicit sensing. Questionnaires are also provided. Different question types such as binary questions (YES/NO), multiple-choice questions (up to four options), and questions that require photo uploads and explanatory text are supported. These questions can be linked together to support a step-by-step questionnaire.

Static/dynamic requests (F3): ParmoSense supports both static and dynamic requests for soliciting contributions to sensing tasks. In PMS for urban environments, many requests are based on geographical information. Static requests place each task as a checkpoint on a map, as shown in Fig. 4(c), which allows participants to easily find the tasks to be performed. For dynamic requests, ParmoSense supports participants by informing them of task requests through notifications as shown in Fig. 5(a). Organizers can generate and request tasks from specific participants, for example, by setting the system so that any person detected entering a certain area is notified. Location information can be obtained using region monitoring technology such as Geofence and iBeacon.

Monetary/non-monetary rewards (F4): ParmoSense supports both monetary and non-monetary rewards to compensate participants for their contributions. The monetary rewards supported are discount coupons that can be used at restaurants, cafes, and so forth, as shown in Fig. 5(b). Since discount coupons are linked with purchasing behavior, they are effective for motivating participation in scenarios such as sightseeing. For non-monetary rewards, ParmoSense supports a gamification mechanism. This includes awarding points for a contribution, competition mechanisms such as comparing a participant's degree of contribution with that of other participants, and virtual level-up elements to reward repeated contributions. Also, depending on the demand for data collection, monetary/non-monetary rewards can be adjusted to influence behavior. For example, a high monetary reward can be set for a task with a low upload rate or a high-priority task.

Feedback by visualization (F5): ParmoSense also supports feedback not included in game features, for example, visualization of own contributions, and data/experience sharing. Visualization of own contributions is of two types: (i) plotting the pins of contributions on a map and (ii) scoring contributions with the non-monetary rewards detailed above. In addition, a data sharing function to share/visualize data such as participants' experiences and what different



Congestion degree
Is convenience store congested?

Nara Taro 1 days ago
SHU tu SHU: In a space of homely atmosphere, you can meet authentic curry.

Nara Taro 1 days ago
REKA: Shops where you can eat delicious Indian food.

Nara Taro 1 days ago
Asa no Ha: Famous store of sweets and foods which are also sold at the department store in Tokyo.

(a) Notification (b) Coupon (c) Experience sharing

Fig. 5. (Color online) Examples of motivating functions. (a) ParmoSense client can send a notification to participants to request a PMS task; (b) ParmoSense can provide a coupon as a monetary incentive; (c) other participants' contributions are shared as a timeline. (The screenshot is extracted from Google Map.)

participants have viewed, heard, or sensed (environmental conditions) is provided. For example, ParmoSense can plot sensing data uploaded by other participants on the map and share them on a timeline as shown in Fig. 5(c).

Editing of collected data (F6): ParmoSense helps organizers to pre-process collected data before detailed analysis. For example, it provides functions such as cleansing unnecessary data, selecting data to be used, and labeling the data. Also, since these data processing functions are designed to protect the original data, the data can be restored at any time.

Browsing of collected data (F7): ParmoSense provides visualization tools for instant and easy confirmation of the collected data. There are many types of tools such as a tool for plotting data collected by all participants or each participant on the map, and a tool for displaying all the data as a list. These tools can be used at any time even while sensing tasks are in progress.

Export of collected data (F8): ParmoSense supports various data output formats. Data analysis data can be output in CSV, JSON, XML, RDB (SQLite), etc., so that analysis can be started immediately. Moreover, when exporting to third-party visualization tools (e.g., Open Street Map,(38) Google Earth,(39) Cesium(40)), it is possible to output in KML (https://developers.google.com/kml/) or GPX (http://www.topografix.com/gpx.asp).

4. Evaluation

ParmoSense has tackled two challenges: the limited support of essential functions in existing PMS platforms (C1) and the difficulty of system construction and operation (C2). In this section, we use the easiness of system construction and operation and the number of functions provided as the metrics, and we compare the performance of ParmoSense with conventional platforms.

4.1 Workload for system operation

We compared the development and operation costs of ParmoSense with those of conventional platforms.(15–23) We used the workload cost for the starting operation of the system (preparation workload) as the metric for comparison. The preparation workload is defined as follows:

Preparation workload (*W*): This is the total time required from the start of developing the PMS system to the start of sensing tasks, which is calculated as the sum of the subtasks [Eq. (1)].

$$W = \sum_{i=1}^{5} w_i \left(0 \le W\right) \tag{1}$$

Here, w_1–w_5 are the relative estimated times required for each subtask, as shown in Table 3. For reference, we defined the time required to install one application from a general application store, e.g., Google Play or AppStore, as $w_x = 1$.

The estimated preparation workload on each platform is shown in Table 4 and on the x-axis of Fig. 6. A filled circle (●) represents the case where the simplest system on each platform was constructed. AWARE,(15) Medusa,(17) OpenDataKit,(22) and GP-Selector(23) can be extended by programming as necessary. However, w_1 increases linearly with the development of functionality.

Table 3
Preparation workload (W) for each subtask.

	Contents of work	Preparation Workload	Done by
w_1	development (programming)	0, 8^{*a}	organizer
w_2	development (GUI editing)	0, 4^{*e}	
w_3	publishing to app store	0, 8^{*b}	
w_4	installing application	0, 1^{*c}, 2^{*d}	participant
w_5	configuring functions	0, 2^{*e}	

*a calculated with LOC of source code. *b calculated based on time for becoming available in general stores. *c installation of single application (reference time). *d installation of multiple application. *e calculated by comparing with w_1, w_3 and w_4 relatively.

Table 4
Breakdown of preparation workload (W) and function score (S) for each platform.

Platforms	Preparation Workload						Function Score
	w_1	w_2	w_3	w_4	w_5	W	S
AWARE [15]	-*a	4	-	2	2	8	4.5
Sensus [16]	-	4	-	1	-	5	3
Medusa [17]	8*a	-	8	1	-	17	3.5
Funf [18]	-	-	-	1	2	3	4.5
MinaQn [19]	-	4	-	-	-	4	3.5
KOKOPIN app [20]	-	4	-	1	-	5	4
Ohmage [21]	-	-	-	2	2	4	5
OpenDataKit [22]	-*a	4	8	2	2	16	4.5
GP-Selector [23]	-*a	4	-	1	-	5	3.5
ParmoSense	-	4	-	1	-	5	8

*a Additional time is required for developing the new functionalities.

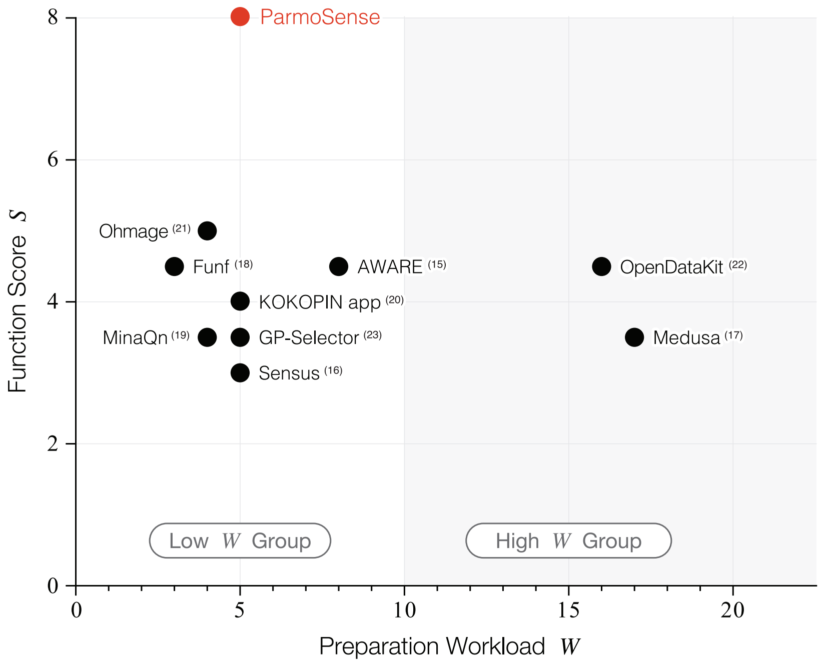


Fig. 6. (Color online) Comparison graph.

ParmoSense belongs to the group with a low preparation workload and can be operated as easily as other platforms in the same group. Also, conventional platforms(15,17,22,23) tend to require a long time for expansion when trying to extend the system functionality. In contrast, ParmoSense is designed to cover the necessary functions in advance, and, thus, although the preparation workload is equivalent to that of other platforms, it achieves higher functionality.

4.2 Variety of functions

We compared the diversity of functions that ParmoSense provides with that of conventional platforms.(15–23) As a metric for comparison, we used the following fulfillment status of functions (function score):

Function score (*S*): Table 1 shows all the functions supported by each existing PMS platform. The score of each function s_1–s_8 is determined by its implementation status (F1–F8) in Table 1, and the function score (S) is calculated as their sum [Eq. (2)].

$$S = \sum_{i=1}^{8} s_i \left(0 \le S \le 8\right) \tag{2}$$

$$s_i = \begin{cases} 1, & \text{if F[i]=}\checkmark \\ 0.5, & \text{if F[i]=}\triangle \\ 0, & \text{otherwise} \end{cases}$$

Here, w_1–w_5 are the relative estimated times required for each task, as shown in Table 3. For reference, we defined the time required to install one application from a general application store, e.g., Google Play or AppStore, as $w_x = 1$.

The estimated function score on each platform is shown in Table 4 and on the y-axis of Fig. 6. While conventional platforms have moderate scores ($S = 4 \pm 1$), ParmoSense has a higher score ($S = 8$) due to its comprehensive support of all the functions provided in existing platforms and its incorporation of motivating functions, which are missing on all existing platforms. Furthermore, the advantage of ParmoSense will be even greater if we consider the effect of combining these functions. For example, ParmoSense can provide combinations of multiple motivating functions, such as gamification, rewards, and interruptions, which cannot be provided on other existing platforms. This may be more effective than using a single motivating strategy/function since more participant preferences are catered for.

5. Case Studies

We conducted 19 case studies with ParmoSense over four years. Prior to the evaluation, we released ParmoSense to general application stores (Google Play, AppStore). We collaborated with various organizers to design and build 19 scenarios and then deployed them via the client application. Members of the general public who had downloaded the application acted as participants.

Our aim in conducting these case studies was twofold. First, we wanted to confirm that our implementations of the sensing, processing, and motivating functions on ParmoSense worked well in real-world PMS tasks. Second, we wanted to determine how effective the functions were in motivating participation and in supporting organizers to collect the required data and extract the required information. Such knowledge would allow us to improve the functions or to recommend additional functions to be included in PMS systems. Table 5 shows the details of each scenario (start date, period, number of participants, and embedded functions). In the following subsections, we discuss the sufficiency of the ParmoSense functions in each scenario.

5.1 Overview of case studies

In this section, we categorize case studies into four groups according to the type of sensing task involved and provide an overview of each group. We then describe how well the ParmoSense functions performed for each scenario.

Urban data collection during workshops (S1–S6): S1–S6 are scenarios for workshop-style events, such as mapping parties,(41–43) which are widely used in organizations such as OpenStreetMap. An overview of each scenario is given as follows:

<u>Scenarios S1 and S2 (Mapping parties):</u> These scenarios targeted the collection of unmapped geographical data. In the event, we collected information on trees (names and positions) in our university campus.

<u>Scenario S3 (FixMyStreet):</u> This scenario targeted the collection of dynamic geographical data such as road breakages, graffiti, and street lamp failures by imitating the mechanism in FixMyStreet.(14)

<u>Scenarios S4–S6 (Urban planning):</u> These scenarios involved surveying existing points of interest, such as local buildings, public facilities, and nature, for urban planning.

Table 5
Overview of case studies.

Scenario No.	Functions								Start date	Period	Number of participants
	sensing functions		motivating functions			processing functions					
	F1 Implicit	F2 Explicit	F3 Request	F4 Reward	F5 Feedback	F6 Editing	F7 Browsing	F8 Export			
S1	✓	✓		✓	✓		✓		Feb. 9th, 2016	0.5 hours	15
S2	✓	✓			✓	✓	✓	✓	Feb. 17th, 2018	1 day	9
S3	✓	✓		✓	✓		✓		June 1st, 2016	0.5 hours	17
S4	✓	✓			✓	✓	✓	✓	Dec. 17th, 2016	2 days	14
S5	✓	✓			✓	✓	✓	✓	June 8th, 2017	1 day	6
S6	✓	✓			✓	✓	✓	✓	June 10th, 2017	1 day	20
S7	✓	✓		✓	✓		✓	✓	June 5th, 2016	2 days	19
S8 (44)	✓	✓	✓		✓	✓	✓	✓	Nov. 22nd, 2016	3 days	14
S9 (45)	✓	✓	✓	✓	✓				Nov. 26th, 2017	1 day	30
S10 (46)	✓	✓	✓	✓	✓				Jul. 29th, 2020	1 day	10
S11	✓	✓	✓	✓	✓				Oct. 5th – Nov. 4th, 2020	1 day/person	108
S12	✓	✓	✓	✓	✓				Jan. 21st, 2016	10 days	12
S13	✓	✓	✓				✓		Apr. 20th, 2017	14 days	83
S14	✓	✓	✓	✓	✓		✓	✓	Mar. 12th, 2016	1 day	10
S15	✓	✓	✓	✓	✓		✓		Nov. 13th, 2016	1 day	48
S16	✓	✓	✓	✓	✓		✓	✓	Feb. 25th, 2017	1 day	25
S17	✓							✓	June 11th, 2017	2 days	18
S18	✓							✓	July 24th, 2017	4 hours	100
S19	✓							✓	Sept. 24th, 2017	6 hours	200

Through the deployment of these scenarios, we obtained the following knowledge:

- Because participants were willing to attend events by themselves, we confirmed that the set of motivating functions on ParmoSense are sufficiently effective to obtain sufficient contributions from the general public.
- The following processing functions worked effectively:
    - Data labeling function
    - Cleansing function for unnecessary data
    - Automatic mosaic function by face recognition
    - Data export function

Urban data collection during sightseeing (S7–S11): S7–S9 are scenarios for PMS during sightseeing, for example, with an information sharing tool for tourists or an urban sensing system mimicking a tourist guide. An overview of each scenario is as follows:

<u>Scenario S7 (Experience sharing):</u> This scenario involved sharing discovered sightseeing spots among a group of tourists. To encourage positive postings among participants, we used points as the motivating function (F4).

<u>Scenario S8 (Tourist guidance):</u> This scenario involved collecting data such as photos and behavior logs from tourists while providing sightseeing information through a virtual tour guide.(44) We provided a data editing function (F6) for organizers to edit collected data after an event.

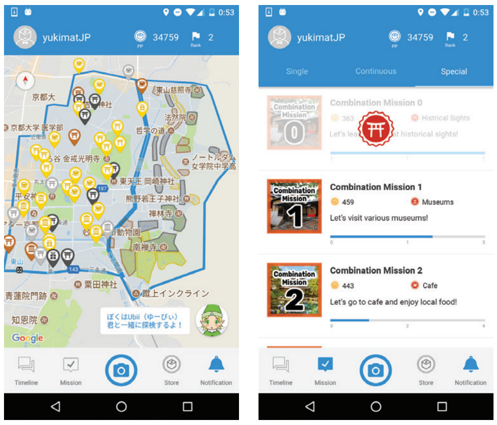
<u>Scenarios S9–S11 (Multitype requests):</u> These scenarios involved the comprehensive collection of environmental data of sightseeing spots by requesting the data in various ways from tourists. To encourage their active contribution, we provided all the available motivating functions such as static/dynamic requests (F3) and points and coupons (F4). Figure 7 shows screenshots of the ParmoSense client, which provides motivating functions including the dynamic control of points based on the demand for sensing tasks. Figure 8 visualizes user trajectories when ParmoSense is used with and without motivating functions (F4). The area marked with dashed lines shows that motivating functions (weighted points) effectively work to change user behaviors by dynamically weighting the points that can be earned. It suggests that the motivating functions in ParmoSense can contribute to resolving the spatio-temporal coverage issue of general PMS systems. A detailed analysis of these case studies (S9 and S10) is provided in our other papers.(45,46)

Through these case studies, we observed the following regarding the effectiveness of the implemented functions for scenarios belonging to this category:

- In the case of sightseeing, motivating functions were essential because tourists participated in the scenario opportunistically.
- Collecting factors, such as points and sharing information with each other, were more effective than competitive factors, such as scores and rankings, for motivating continuous participation because sightseeing was the primary purpose of the participants and collecting factors was more relevant.

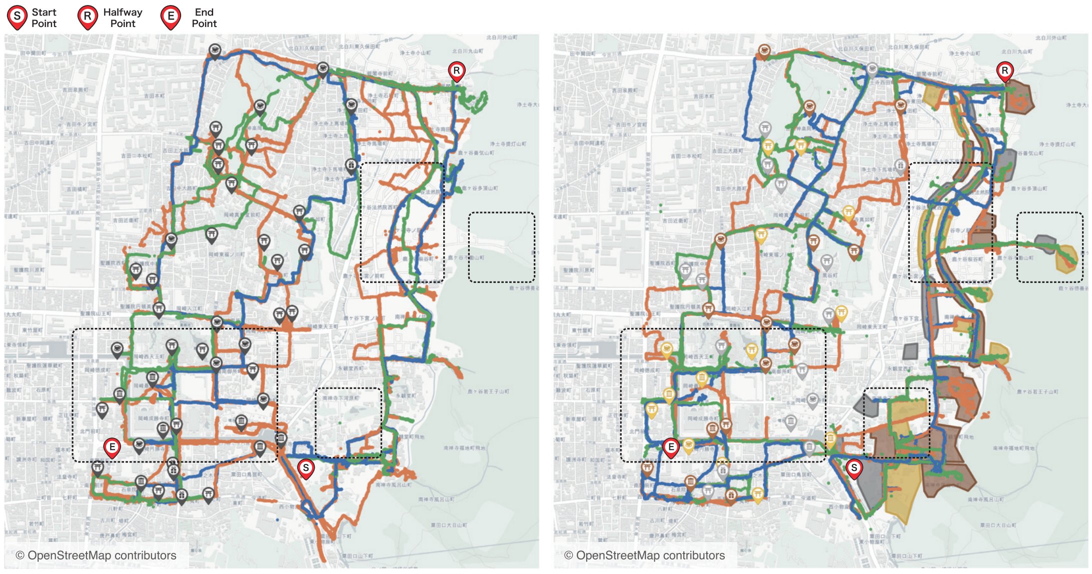
Urban data collection in daily life (S12 and S13): The data suitable for collecting by PMS are often continuous and long term, because PMS is a sustainable sensing mechanism to realize comprehensive spatio-temporal data collection without any infrastructure due to the use of the many mobile devices dispersed in a city. S12 and S13 were scenarios that involved collecting such long-term and continuous data. An overview of each scenario is as follows:

(a) Participant who is taking photo in Scenario S9



(b) Screenshots of ParmoSense Client in Scenario S9

Fig. 7. (Color online) Photo of participant and screenshots of ParmoSense client in Scenario S9.(45) (The screenshot is extracted from Google Map.)



(a) with non-gamified system (b) with gamified system (weighted points)

Fig. 8. (Color online) User trajectories with/without motivating functions in Scenario S9.(45)

<u>Scenario S12 (Static requests):</u> This scenario involved periodically collecting information that changes day by day, such as bulletin board information at a facility and the temperature of a location. To encourage participation, we set a limit on the number of participant contributions resulting from static requests (F3) that would be accepted, which is similar to the audition mechanism.(17)

<u>Scenario S13 (Dynamic requests):</u> This scenario involved conducting questionnaires linked to location information by interrupting participants. Push notifications were used and participant location and behavior were taken into consideration to provide dynamic requests (F3). No maps, rewards, or visualizations were included.

Through the case studies involving these scenarios, we concluded the following regarding the effectiveness of ParmoSense functions intended for urban data collection in daily life:

- In the case of S12, due to the use of static requests through placing pins on a map, the achievement of sensing depended on the active and continuous contribution of participants themselves. The following functions worked efficiently when motivating contribution:
    - ◦ Virtual rewards (e.g., points, rankings, levels)
    - ◦ Monetary incentives
    - ◦ Visualization of contributions of the participant and other participants
- Restricting the number of contributions for static requests was effective for encouraging continuous participation because it prevented point inflation, but it also caused some participants to leave.
- In the case of S13, we found that the contribution of participants increased when the timing of requests was based on their location and behavior; for example, participants were more likely to respond to requests if the location of the sensing task was near their present location.

Investigation of human behavior during events (S14–S19): ParmoSense can collect not only urban environmental data, but also data of people in a city. S14–S19 were scenarios created for investigating human behavior during various situations such as events, daily life, and sightseeing. An overview of each scenario is as follows:

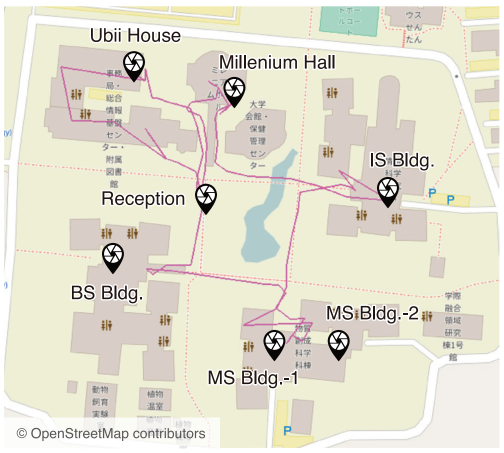
<u>Scenarios S14–S16 (Stamp rally):</u> These scenarios involved investigating the behavior of people participating in an electronic "stamp rally", which is a process where visitors are instructed to visit certain locations or events.(47) A physical (electronic) stamp is situated in a predetermined location, and when a visitor arrives, a sheet (app) is stamped. A reward is given on the basis of the number of stamps accumulated. Non-monetary/monetary rewards include coupons, prizes, points, and rankings (F4). We provided a data browsing function for visualizing participant behavior (F7). Figure 9 shows examples of visualized participant behavior.

<u>Scenario S17 (Sightseeing):</u> This scenario involved collecting participant movement data linked with location information to investigate participant behavior during sightseeing. To continuously sense participant behavior, we used an implicit sensing function that ran in the background (F1).

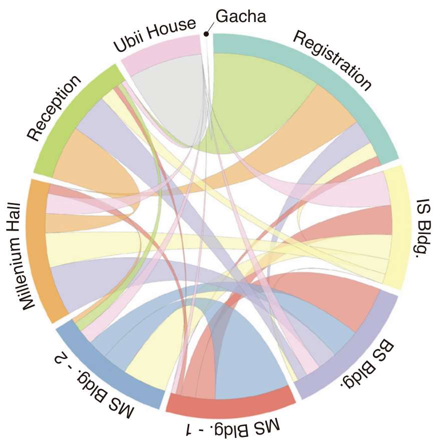
<u>Scenarios S18 and S19 (W/external sensor):</u> These scenarios were used to collect participant movement and heart rate data linked to location information to construct a database for human behavior analysis. We added a function to connect a wearable sensor to external sensors (F1) via BLE for the collection of data.

Through these case studies, we learned the following about PMS for human behavior investigation:

- To analyze the behavior of a person, it is necessary to visualize behavior logs in various ways. We found that the functions provided by ParmoSense such as GPS tracing and chord diagram listings worked effectively.
- Since these scenarios were aimed at sensing the effect of a specific function on human behavior, it was necessary to suppress the interference of elements except for the function to be evaluated. We confirmed that ParmoSense can minimize the number of unnecessary functions because of its module-based design.
- It is sometimes necessary to collect data with high frequency over a long period (S18 and S19 were scenarios that involved acquiring nine-axis sensor data at 100 Hz over several hours). In this case, we found that background functions such as continuous data acquisition worked efficiently.



(a) GPS trace of participant



(b) Chord diagram of behavior

Fig. 9. (Color online) Results of human behavior investigation in Scenario S15.

5.2 Survey and discussion

In the following sections, we summarize and discuss the results of the subjective surveys held in each case study. We asked each participant and organizer questions about the usability and performance of ParmoSense.

5.2.1 Overview

Participants in nine scenarios (S2, S4–S6, S9, S13–S16) were given a questionnaire. These scenarios involved the participation of many ordinary citizens and included at least one motivating function. The total number of participants who answered the questionnaire was 201. About 70% of the participants were in their 20s; however, various age groups ranging from 10–19 to 80 and above were targeted. The attributes of the participants are listed below:

- Men and women of all ages living in the local area (S2, S4, S15, and S16)
- Undergraduate students in their 20s majoring in architectural engineering (unfamiliar with information science) (S5, S6)
- Graduate students majoring in information science (S9, S13)

We also distributed questionnaires to the organizers of five scenarios (S2, S4–S6, S9). The skills of the organizers varied from those with high IT skills such as IT engineers to those with low IT skills such as students majoring in fields other than IT. The attributes of the organizers are listed below:

- Employee of regional public facility (S2)
- IT engineer (S4)
- Undergraduate students majoring in architectural engineering (S5, S6)
- Graduate students majoring in information science (S9)

5.2.2 Survey of participants

To evaluate the usability of the ParmoSense client, we asked participants the questions listed in Table 6. We asked Q1 (“Was ParmoSense client easy to use?”) with a five-point Likert scale (5: very easy to use – 1: very hard to use), and then asked for the reason in Q2. The total number of answers was 196. The breakdown of answers is shown in Table 6. The average score was 3.4, and about 37.5% of participants answered “easy to use” (answer of 4 or 5).

Participants who answered “easy to use” gave the following comments:

- “We could operate intuitively because of the simple design of the application.” (S4, S6, S16)
- “The map visualization function is helpful for understanding the collected data with position information.” (S4–S6)

Those who answered “hard to use” (answer of 1 or 2) gave the following comments:

- “It was difficult to use the ParmoSense application because I was not used to smartphones.” (S4, S15)
- “The ParmoSense application was not suitable for long-term use because it consumed battery more than expected.” (S4–S6)
- “Unused functions should be invisible.” (S13)

Overall, we found that using ParmoSense was easy for participants who use smartphones on a daily basis regardless of age. Some of the elderly persons who attended local events felt that it was difficult to use a smartphone. However, this is not a particular problem of ParmoSense. As the range of use cases diversifies, the information that should be displayed on the screen will also be different. To resolve this issue without programming skills, it is necessary to modularize the screen configuration of the application and also to enable the screen layout to be designed using Scenario Tools.

Next, we asked Q3 (“Was the event using ParmoSense fun?”) with a five-point Likert scale (5: very fun – 1: not fun at all) in the target scenarios except for S13. Additionally, we also asked Q4 to solicit free responses from participants on how motivating functions such as the reward and visualization affected their sensing behavior. The total number of answers was 103. The breakdown of answers is shown in Table 6. The average score from the responses was 4.2 and about 82.5% of participants answered “fun” (answer of 4 or 5).

Table 6
Questionnaire items for participants and non-participants (S2, S4–S6, S9, S13–S16).

Item No.	Questionnaire Detail	Answer					
		1 (disagree)	2	3	4	5 (agree)	Average
Q1	Was ParmoSense Client easy touse? (S2, S4–S6, S9S13–S16)	8 (5.0%)	19 (11.9%)	73 (45.6%)	35 (21.9%)	22 (15.6%)	3.4
Q2	Give the reason for your answer to Q1? (S2, S4–S6, S9, S13–S16)	(Open-ended question)					-
Q3	Was the event using ParmoSense fun? (except S13)	- (0.0%)	4 (3.9%)	14 (13.6%)	42 (40.8%)	43 (41.7%)	4.2
Q4	Give the reason for your answer to Q3? (except S13)	(Open-ended question)					-
Q5	What factors would encourage you to participate in experiments? (Non-participants of S13)	(Open-ended question)					-

For Q4, we obtained the following comments:

- "It is pleasant to see the behavior of other people and other groups by checking the pins on the map and the timeline in real time." (S4, S12)
- "The ranking function and level function make it fun and encouraging." (S4, S5, S9)
- "This application increased the fun of the stamp rally." (S15, S16)

5.2.3 Survey of non-participants

We also provided questionnaires to non-participants in the free participation scenario (S13) to explore the reasons why they did not participate in the sensing tasks.

In scenario S13, 35 of 118 candidates did not respond to the request to participate in our experiment. We asked these 35 non-participants the open-ended question Q5 ("What factors would encourage you to participate in experiments?"). The factors that non-participants felt would promote participation were related to ease of participation (17 people), battery consumption concerns (10 people), rewards (13 people), and benefit or convenience (12 people). Furthermore, four people were concerned about privacy and commented on the need to disclose personal information such as an e-mail address and the inability to be anonymous.

Almost half of the non-participants pointed out that the number of steps they needed to take to participate was inconvenient. To minimize the steps required to participate in multiple scenarios, we first required participants to install the ParmoSense client. After installing the application, participants read a QR code for each scenario to participate in that particular scenario. However, people who participated in only one scenario felt that this was complicated and that having a single, pre-configured application for a specific scenario might be easier.

Also, to make the registration and login process easier, we adopted Google Authentication (using a Gmail address) and supported auto-login. This is a standard method used in many applications. However, some participants felt that our application could collect private data such as their e-mail address. In the future, we aim to address this by introducing an anonymous participation mode that does not require participant registration.

5.2.4 Survey of organizers

To evaluate the effect of ParmoSense from the view of the organizers and the usability of its processing functions, we conducted a questionnaire with the organizers. The questions for the organizers are listed in Table 7. Q6–Q11 were answered with a four-point Likert scale (4: strongly agree – 1: strongly disagree) and Q12 was a free-response question.

To evaluate the burden on the organizers, referred to as the distributing phase in Fig. 1, we asked Q6 (the ease of creating and distributing scenarios). Two people answered "4" and the other three answered "3," giving an average score of 3.4. Although the IT skills of the organizers varied considerably, all of them felt that ParmoSense was easy to use for PMS.

Next, we asked Q7 (whether organizers were able to collect the desired data) and Q8 (the performance of the motivating function) to obtain feedback from organizers about the sensing phase (Fig. 1). Four organizers answered "4" and one answered "3" to Q7, while two answered

Table 7
Questionnaire items for organizers (S2, S4–S6, S9).

Item No.	Questionnaire Detail	Answer				
		1 (disagree)	2	3	4 (agree)	Average
Q6	Was ParmoSense easy to introduce to the event?	-	-	3	2	3.4
Q7	Could you collect the desired data?	-	-	1	4	3.8
Q8	Were participants' motivation for attending to events improved by using ParmoSense? (e.g., ranking, coupons)	-	-	3	2	3.4
Q9	Was Data tools easy to use?	-	1	2	2	3.2
Q10	Were the data outputted by the data output function easy to use secondary diversion and data processing?	-	1	3	1	3.0
Q11	Do you want to use ParmoSense again in future similar events?	-	-	2	3	3.6
Q12	Why did you think so about the answer to Q11?	(Open-ended question)				-

"4" and the other three answered "3" to Q8. Consequently, we concluded that ParmoSense allows organizers to collect the desired data and motivate participants to contribute to continuous sensing.

To evaluate the processing phase in Fig. 1, we asked the organizers Q9 (usability of Data Tools) and Q10 (availability of output data). Two organizers answered "4", two answered "3", and one answered "2" for Q9, while one organizer answered "4", three answered "3", and one answered "2" for Q10. The average scores were 3.2 and 3.0 for Q9 and Q10, respectively. From these results, we confirmed that Data Tools of ParmoSense works well.

Additionally, we interviewed the organizers who answered "2" in Q9 and Q10. The organizer whose answer for Q9 was "2" said that unexpected data was downloaded when data was exported for each tag after tagging the data. Therefore, as an additional question, we asked whether ParmoSense would be easy to use if data could be successfully downloaded, and the organizer thought it would be. This shows that although it is necessary to improve the flexibility of the export function, it appears that the usability of the Web editor meets specific service standards. The organizer whose answer for Q10 was "2" said that during the web visualization, the operation became slow and the PC froze many times. About 700 photos were collected in this organizer's event. This amount of data was too large for all photos to be displayed at once. In the future, it will be necessary to set an upper limit on the number of photos that can be displayed or reduce the image size according to the number of images before uploading.

Finally, three organizers answered "4" and two answered "3" to Q11 (whether they would use ParmoSense again). When we asked the reason (Q12), the following answers were obtained:

- "It can find places that the participants are interested in."
- "It can easily collect data with location information and can edit detailed information later. In addition, by visualizing the data, participants can intuitively understand how the data is used."

From these results, we found that ParmoSense is useful for human behavior analysis and for feeding back results to participants and that it is easy to edit data.

6. Conclusion

Demand for the assessment of urban environments through PMS systems exists not only in the information science field, but also in a wide range of areas, such as urban planning and public services. In such areas, organizers are not always familiar with information technology. Therefore, ParmoSense was designed as a useful tool for collecting urban environmental information easily and flexibly regardless of the IT skills of the organizer. ParmoSense allows organizers to construct, distribute, and introduce various types of PMS system for different purposes by modularizing functions and describing combinations and settings as “scenarios.” In PMS systems that require ordinary citizens to collect data, motivating participants is also important for their continuous operation. Thus, unlike conventional platforms, ParmoSense incorporates several motivating functions.

Through a performance comparison with existing PMS platforms from the perspectives of the variety of functions and the preparation workload, we confirmed that ParmoSense has the best cost performance. In addition, through 19 case studies over four years, we confirmed that ParmoSense can be used flexibly with various sensing tasks, motivation methods, and data processing methods. Moreover, ParmoSense can reduce the burden on organizers and participants during its system operations. On the other hand, we also found that ParmoSense is not a “magic bullet” solution that can be used in every situation. The privacy and security of ParmoSense should be improved by allowing organizers to control privacy and security settings according to the purpose of the sensing. In addition, to employ ParmoSense in scenarios that involve a larger number of participants, robustness and scalability should be guaranteed through distributed processing and the prediction of score updates. We hope that our work will help establish a smart city in the era of Society 5.0 by encouraging the co-creation of government, company, academia, and citizens.

Acknowledgments

This research was funded by JST PRESTO (JPMJPR2039) and JSPS KAKENHI (JP19K24345 and JP19H01139).

References

1 J. A Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B Srivastava: UCLA: Center for Embedded Network Sensing (2006) 1. https://escholarship.org/uc/item/19h777qd
2 A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson: Proc. 2nd Annu. Int. Workshop Wireless Internet (ACM, 2006) 18. https://doi.org/10.1145/1234161.1234179
3 E. Paulos, R. Honicky, and B. Hooker: Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City (IGI Global, Pennsylvania, 2008) Chap. 28. https://doi.org/10.4018/978-1-60566-152-0.ch028
4 S. Morishita, S. Maenaka, N. Daichi, M. Tamai, K. Yasumoto, T. Fukukura, and K. Sato: Proc. 2015 ACM Int. Joint Conf. Pervasive and Ubiquitous Computing (ACM, 2015) 695–705. https://doi.org/10.1145/2750858.2804273
5 K. Konis and M. Annavaram: Build. Environ. **118** (2017) 1. https://doi.org/10.1016/j.buildenv.2017.03.025
6 C. R. D. Rolt, R. Montanari, M. L. Brocardo, L. Foschini, and J. D. S. Dias: Proc. 2016 IEEE Symp. Computers and Communication (IEEE, 2016) 999–1005. https://doi.org/10.1109/ISCC.2016.7543867
7 S. Heggen: Proc. 2012 ACM Conf. Ubiquitous Computing (ACM, 2012) 552–555. https://doi.org/10.1145/2370216.2370307

8 K.-L. Ong, S. Leao, and A. Krezel: Int. J. Pervasive Comput. and Commun. **10** (2014) 419. https://doi.org/10.1108/IJPCC-04-2014-0030
9 Y. Matsuda and I. Arai: Adj. Proc. 2014 ACM Int. Joint Conf. Pervasive and Ubiquitous Computing (ACM, 2014) 115–118. https://doi.org/10.1145/2638728.2638797
10 E. Kanjo: Mobile Networks and Appl. **15** (2010) 562. https://doi.org/10.1007/s11036-009-0217-y
11 N. Maisonneuve, M. Stevens, and B. Ochab: Inf. Polity **15** (2010) 51. https://doi.org/10.3233/IP-2010-0200
12 D. Mendez, A. J. Perez, M. Labrador, and J. J. Marron: Proc. 2011 IEEE Int. Conf. Pervasive Computing and Communications Workshops (IEEE, 2011) 344–347. https://doi.org/10.1109/PERCOMW.2011.5766902
13 Y. Zheng, F. Liu, and H.-P. Hsieh: Proc. 19th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (ACM, 2013) 1436–1444. https://doi.org/10.1145/2487575.2488188
14 FixMyStreet Platform: http://fixmystreet.org/ (accessed April 2022).
15 D. Ferreira, V. Kostakos, and A. K. Dey: Front. ICT **2** (2015) 1. https://doi.org/10.3389/fict.2015.00006
16 H. Xiong, Y. Huang, L. E. Barnes, and M. S. Gerber: Proc. 2016 ACM Int. Joint Conf. Pervasive and Ubiquitous Computing (ACM, 2016) 415–426. https://doi.org/10.1145/2971648.2971711
17 M.-R. Ra, B. Liu, T. F. L. Porta, and R. Govindan: Proc. 10th Int. Conf. Mobile Systems, Applications, and Services (ACM, 2012) 337–350. https://doi.org/10.1145/2307636.2307668
18 N. Aharony, W. Pan, C. Ip, I. Khayal, and A. Pentland: Pervasive Mob. Comput. **7** (2011) 643. https://doi.org/10.1016/j.pmcj.2011.09.004
19 M. Sakamura, T. Ito, H. Tokuda, T. Yonezawa, and J. Nakazawa: Adj. Proc. 2015 ACM Int. Joint Conf. Pervasive and Ubiquitous Computing and Proc. 2015 ACM Int. Symp. Wearable Computers (ACM, 2015) 1607–1614. https://doi.org/10.1145/2800835.2801632
20 M. Mishima, T. Matsumoto, S. Takano, and O. Matsuda: Proc. 2nd ACM Int. Workshop Multimedia Analysis for Ecological Data (ACM, 2013) 35–40. https://doi.org/10.1145/2509896.2509901
21 H. Tangmunarunkit, C. K. Hsieh, B. Longstaff, S. Nolen, J. Jenkins, C. Ketcham, J. Selsky, F. Alquaddoomi, D. George, J. Kang, Z. Khalapyan, J. Ooms, N. Ramanathan, and D. Estrin: ACM Trans. Intell. Syst. Technol. **6** (2015) 1. https://doi.org/10.1145/2717318
22 W. Brunette, M. Sundt, N. Dell, R. Chaudhri, N. Breit, and G. Borriello: Proc. 14th Workshop Mobile Computing Systems and Applications (ACM, 2013) 1–6. https://doi.org/10.1145/2444776.2444790
23 J. Wang, Y. Wang, L. Wang, and Y. He: World Wide Web **21** (2017) 759. https://doi.org/10.1007/s11280-017-0480-y
24 Y. Arakawa and Y. Matsuda: J. Inf. Process. **24** (2016) 31. http://id.nii.ac.jp/1001/00147383/
25 Gamification Pitfalls: Badge Fatigue and Loyalty Backlash: http://www.gamification.co/2012/09/12/gamification-pitfalls-badge-fatigue-and-loyalty-backlash/ (accessed April 2022).
26 F. Restuccia, S. K. Das, and J. Payton: ACM Trans. Sens. Netw. **12** (2015) 1. https://doi.org/10.1145/2888398
27 L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador: Proc. 2012 IEEE Int. Conf. Pervasive Computing and Communications (IEEE, 2012) 103–108. https://doi.org/10.1109/PerCom.2012.6199855
28 H. Gao, C. H. Liu, W. Wang, J. Zhao, Z. Song, X. Su, J. Crowcroft, and K. K. Leung: IEEE Commun. Surv. Tutorials **17** (2015) 918. https://doi.org/10.1109/COMST.2014.2387836
29 R. I. Ogie: Human-centric Computing Inf. Sci. **6** (2016) 1. https://doi.org/10.1186/s13673-016-0080-3
30 I. Krontiris and A. Albers: Int. J. Parallel Emergent Distrib. Syst. **27** (2012) 317. https://doi.org/10.1080/17445760.2012.686170
31 G. Zichermann and C. Cunningham: Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps (O'Reilly Media, California, 2011). https://dl.acm.org/doi/10.5555/2073550
32 S. Deterding, D. Dixon, R. Khaled, and L. Nacke: Proc. 15th Inte. Academic MindTrek Conf.: Envisioning Future Media Environments (ACM, 2011) 9–15. https://doi.org/10.1145/2181037.2181040
33 F. Groh: Proc. the 4th Seminar on Research Trends in Media Informatics (Institute of Media Informatics Ulm University, 2012) 39–46. http://hubscher.org/roland/courses/hf765/readings/Groh_2012.pdf
34 Y. Wang, X. Jia, Q. Jin, and J. Ma: Concurrency and Comput.: Pract. Experience **29** (2016) e3789. https://doi.org/10.1002/cpe.3789
35 Y. Ueyama, M. Tamai, Y. Arakawa, and K. Yasumoto: Proc. 2014 IEEE Int. Conf. Pervasive Computing and Communication Workshops (IEEE, 2014) 98–103. https://doi.org/10.1109/PerComW.2014.6815172
36 funf | Open Sensing Framework: http://funf.org/ (accessed April 2022).
37 MQTT: The Standard for IoT Messaging: http://mqtt.org/ (accessed April 2022).
38 OpenStreetMap: http://www.openstreetmap.org/ (accessed April 2022).
39 Google Earth: https://www.google.com/earth/ (accessed April 2022).
40 CesiumJS – Cesium: https://cesium.com/cesiumjs/ (accessed April 2022).
41 M. Haklay and P. Weber: IEEE Pervasive Comput. **7** (2008) 12. https://doi.org/10.1109/MPRV.2008.80

42 D. Hristova, G. Quattrone, A. Mashhadi, and L. Capra: Proc. Int. AAAI Conf. Web and Social Media (AAAI, 2013) 234–243. https://ojs.aaai.org/index.php/ICWSM/article/view/14416

43 P. Mooney, M. Minghini, and F. S.-Jones: Int. J. Spatial Data Infrastruct. Res. **10** (2015) 138. https://ijsdir.sadl.kuleuven.be/index.php/ijsdir/article/view/395

44 M. Hidaka, Y. Kanaya, S. Kawanaka, Y. Matsuda, Y. Nakamura, H. Suwa, M. Fujimoto, Y. Arakawa, and K. Yasumoto: Smart Cities **3** (2020) 212. https://doi.org/10.3390/smartcities3020013

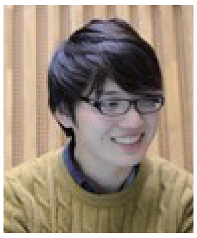
45 S. Kawanaka, Y. Matsuda, H. Suwa, M. Fujimoto, Y. Arakawa, and K. Yasumoto: Smart Cities **3** (2020) 736. https://doi.org/10.3390/smartcities3030037

46 S. Kawanaka, J. Miehle, Y. Matsuda, H. Suwa, K. Yasumoto, and M. Wolfgang: Proc. 17th EAI Int. Conf. Mobile and Ubiquitous Systems: Computing, Networking and Services (ACM, 2020) 458–463. https://doi.org/10.1145/3448891.3448957

47 Rally: https://rallyapp.jp/ (accessed April 2022).

About the Authors

Yuki Matsuda received his B.E. degree from the Advanced Course of Mechanical and Electronic System Engineering, National Institute of Technology, Akashi College, Japan, in 2015 and his M.E. and Ph.D. degrees from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2016 and 2019, respectively. He is currently an assistant professor at the Graduate School of Science and Technology, Nara Institute of Science and Technology, Japan. His current research interests include urban sensing, internet of things, ubiquitous computing, and affective computing. He is a member of IEEE, IEICE, and IPSJ. (yukimat@is.naist.jp)



Shogo Kawanaka received his B.E. degree from the National Institute of Technology, Nara College, Japan in 2015 and his M.E. and Ph.D. degrees from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2018 and 2021, respectively. He is currently working at CyberAgent, Inc. His research interests include participatory urban sensing and vehicle behavior sensing. (kawanaka.shogo.kp1@is.naist.jp)

Hirohiko Suwa has been an assistant professor with Ubiquitous Computing Systems Laboratory at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, since 2014. His current research interests are social informatics, real data mining, and social network analysis. He is a member of IEEE, IEICE, and IPSJ. (h-suwa@is.naist.jp)

Yutaka Arakawa received his B.E., M.E., and Ph.D. degrees from Keio University, Japan, in 2001, 2003, and 2006, respectively. From 2006 to 2013, he was an assistant professor at Keio University and Kyushu University, and from 2013 to 2019, he was an associate professor at Nara Institute of Science and Technology. Since 2019, he has been a professor at the Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University. His current research interests include human activity recognition, behavior change support systems, and location-based information systems. He is a member of ACM, IPSJ, and IEICE. (arakawa@ait.kyushu-u.ac.jp)

Keiichi Yasumoto received his B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1991, 1993, and 1996, respectively. He is currently a professor at the Graduate School of Science and Technology, Nara Institute of Science and Technology. His research interests include distributed systems, mobile computing, and ubiquitous computing. He is a member of ACM, IEEE, IPSJ, SICE, and IEICE. (yasumoto@is.naist.jp)