geosciences

MDPI

Article

A Web GIS Framework for Participatory Sensing Service: An Open Source-Based Implementation

Yu Nakayama [1,*], Kazuhiko Nakamura [2], Hitoshi Saito [3] and Rui Fukumoto [4]

[1] Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, Japan
[2] Center for Spatial Information Science, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, Japan; k_nakamura@csis.u-tokyo.ac.jp
[3] College of Economics, Kanto Gakuin University, 1-50-1 Mutsuura-higashi, Kanazawa-ku, Yokohama, Japan; hsaito@kanto-gakuin.ac.jp
[4] Open Concierge, #307 3-21-5 Kandasakumacho, Chiyoda-ku, Tokyo, Japan; rui@openconcierge.org
* Correspondence: yu-nakayama@mcl.iis.u-tokyo.ac.jp; Tel.: +81-03-5452-6268

Academic Editors: Ruiliang Pu and Jesus Martinez-Frias
Received: 1 March 2017; Accepted: 6 April 2017; Published: 10 April 2017

Abstract: Participatory sensing is the process in which individuals or communities collect and analyze systematic data using mobile phones and cloud services. To efficiently develop participatory sensing services, some server-side technologies have been proposed. Although they provide a good platform for participatory sensing, they are not optimized for spatial data management and processing. For the purpose of spatial data collection and management, many web GIS approaches have been studied. However, they still have not focused on the optimal framework for participatory sensing services. This paper presents a web GIS framework for participatory sensing service (FPSS). The proposed FPSS enables an integrated deployment of spatial data capture, storage, and data management functions. In various types of participatory sensing experiments, users can collect and manage spatial data in a unified manner. This feature is realized by the optimized system architecture and use case based on the general requirements for participatory sensing. We developed an open source GIS-based implementation of the proposed framework, which can overcome financial difficulties that are one of the major problems of deploying sensing experiments. We confirmed with the prototype that participatory sensing experiments can be performed efficiently with the proposed FPSS.

Keywords: participatory sensing; web GIS; crowdsensing; urban sensing

1. Introduction

There are millions of mobile phones in use around the world. Mobile phone sensing is an emerging area of research interest [1]. Today's mobile phones have a number of embedded sensors, including an accelerometer, digital compass, gyroscope, GPS, microphone, and camera. In addition, by affixing a sensory device, mobile sensing provides the opportunity to collect various types of information. Thus, mobile phone sensing holds great promise for a wide range of research fields, such as health monitoring, human behavior, social interaction, traffic monitoring, commerce, and environment monitoring [2]. There are two mobile phone sensing paradigms: participatory sensing and opportunistic sensing. They can be discussed in the literature of mobile crowdsensing [3,4]. In participatory sensing, mobile phone users actively participate in sensing activities. In opportunistic sensing, sensing activities are automated, without the users' involvement. This paper focuses on participatory sensing, which enables public and professional users to gather, analyze and share local knowledge [5].

The common architectural components for participatory sensing applications include mobile device data capture, personal data stream storage, and leveraged data processing [6]. To efficiently develop participatory sensing services, some server-side technologies have been proposed [7,8].

Geosciences **2017**, *7*, 22; doi:10.3390/geosciences7020022 www.mdpi.com/journal/geosciences

Although they provide a good platform for participatory sensing, they are not optimized for the spatial data management and processing. If the spatiality of collected data has important meaning, then the data need to be managed as GIS data. For the purpose of spatial data collection and management, many web GIS approaches have been studied [9–13]. However, they still have not focused on the optimal framework for participatory sensing services. Participatory sensing has not been sufficiently discussed in GIS literature, and vice versa. Moreover, the existing solutions have not given a general framework for providing participatory sensing services based on the comprehensive analysis of the system requirements.

Therefore, this paper defines the general web GIS framework for participatory sensing service (FPSS) based on the comprehensive analysis of the requirements. The proposed FPSS enables an integrated deployment of a platform for participatory sensing services that provides spatial data capture, storage, and data management functions. In various types of participatory sensing experiments, users can collect and manage spatial data in a unified manner. To show the validity of the proposed framework, we introduce an open source-based implementation of the FPSS. The proposed concept can be implemented in other ways, e.g., the spatial optimizations of Apache Spark Streaming can be employed to improve the scalability of real-time processing.

This paper is organized as follows. Section 2 describes the related work. In Section 3, we discuss and define the requirements for the data structure and workflow of the FPSS. Based on the requirements, we describe the architecture and the use case of the proposed FPSS in Section 4. Then, Section 5 introduces an open source-based implementation of the FPSS based on [13]. The effectiveness of the proposed FPSS is confirmed with the application study described in Section 6. We conclude this work in Section 7.

2. Related Work

2.1. Participatory Sensing Framework

To efficiently develop participatory sensing services, some frameworks have been developed. To provide services via a cloud computing system, Sensing-as-a-Service (S^2aaS) was introduced [7]. The requirements for an S^2aaS cloud are (1) supporting various applications on different smartphone platforms; (2) energy-efficiency; and (3) incentive mechanisms that attract mobile users to engage in sensing activities. A middleware framework was developed to build large-scale mobile phone sensing test beds [8]. This framework gives researchers a subset of available mobile devices on which to deploy experiments, and gives users fine-grained control over what sensor information they want to share to guard their privacy. An experimental result on top of the Android platform was reported. To help in the data collection process of participatory sensing, a framework for data quality and feedback was proposed [14]. In this work, they proposed a model of sensing behavior of individuals and appropriate feedback mechanisms. As an evolution of participatory sensing, mobile crowd sensing (MCS) is proposed as a new sensing paradigm, which leverages the power of citizens for large-scale sensing [15]. MCS has two features: (1) involving both implicit and explicit participation; and (2) collecting data from mobile social networks and mobile sensing. An integrated framework was proposed to enable cloud-based and people-centric applications on mobile devices [16]. The proposed approach exploited cloud computing to provide a delivery model targeted to mobile applications. It achieved significant energy saving in the mobile device, but it does not focus on participatory sensing.

In addition, privacy is one of the major problems in participatory sensing systems. In [17], they identified the unique challenges assuming an un-trusted central data server model, and proposed a privacy-aware framework. An anonymity-preserving reputation framework called IncogniSense was proposed to assess the user contributions without privacy threats [18]. It utilizes periodic pseudonyms generated using blind signature and relies on reputation transfer between them. The ParticipAct [19,20] is based on an Android framework [21] for simplifying the quick design of mobile sensing campaigns/projects and offering app developers a set of functions to quickly design their own mobile sensing services.

2.2. Crowdsensing Framework

Mobile crowdsensing is a strong paradigm for practically the same problem as participatory sensing [3,4,22,23]. The crowdsensing concepts to location tracking and collection of spatial data have been proposed in [24,25]. It was pointed out that the barriers for engaging people in crowdsensing are the impact on their battery lifetime and phone usage habits in [24]. To address these requirements, they proposed an energy-efficient application for measuring the location and signal strength data. For vehicular mobile crowd sensing systems, a heterogeneous sensing vehicle selection (HVS) method for the collection of comprehensive tempo-spatial sensing data was proposed in [25]. In this method, a utility function is determined to estimate the sensing capacity of the vehicles based on the spatial distribution and sensing interfaces of sensing vehicles and the coverage of collected data.

2.3. Web GIS Approach

For the purpose of spatial data collection and management, many web GIS approaches have been proposed. An open source GIS-based Java web application that automatically writes HTML and JavaScript code was developed [9]. WAMP-based and Apache Tomcat-based web GIS frameworks were developed and compared [10]. A web browser-based system for online fatal disease mapping was developed and it enables spatial visualization, detection and monitoring of incidence in real-time, without the need for any additional software [11]. An open source-based web GIS application was proposed to share the information and spatial data, allowing users with limited GIS knowledge to access the information customized for specific applications [12]. A field survey assisting system was developed in [13]. OWGIS version 2.0 is a Java and JavaScript application that builds easily configurable Web GIS sites [26]. This OWGIS generates mobile interfaces based on HTML5 technology, and can be used to create mobile applications.

However, these works have not discussed the optimal framework for participatory sensing services.

2.4. Contribution

The contribution of this paper is defining the general framework for providing participatory sensing services based on the comprehensive analysis of the system requirements. The proposed FPSS provides an efficient platform for participatory sensing services, which enables an integrated deployment of spatial data capture, storage, and management functions. The related existing solutions have not provided such a general framework. The goal of the FPSS is for mobile sensing organizers to easily launch sensing projects, not for app developers as [19,20]. Moreover, the FPSS can be used across multiple scales; an individual, a group, or a community [1], not only crowd sensing, and can coexist with privacy-aware frameworks.

3. Requirement

First, we explain the system requirements for the FPSS.

3.1. Definition

First, we define the participatory sensing experiments on which this work focuses. A sensing experiment is performed by an individual, a group, or a community [1]. The features of sensing experiments, including study area, survey items, or the purpose, are not limited to specific types. The spatial data type is point; participants record the measured data along with the location where they are obtained. The FPSS needs to provide any sensing experiment service that satisfies this definition.

3.2. Data Structure

To efficiently record data in a sensing experiment, we define that the recorded information consists of two factors: basic information about the experiment which is called metadata and spatial data obtained for each recording point in the field. This definition is depicted in Figure 1.

Metadata are uniquely recorded for an experiment and contain information about the experiment itself. They consist of title, date, study area, item, keywords, notes, and basemaps.

Spatial data are recorded along with the location and the values for the items linked; for example, the species, height, and an image file of a tree at each recording point. It is usual to obtain multiple spatial data for an experiment.

The survey items can be set as any data type, because the proposed framework should support various types of participatory sensing experiments. Thus, the table structure in the database should be optimized for the above data structure.

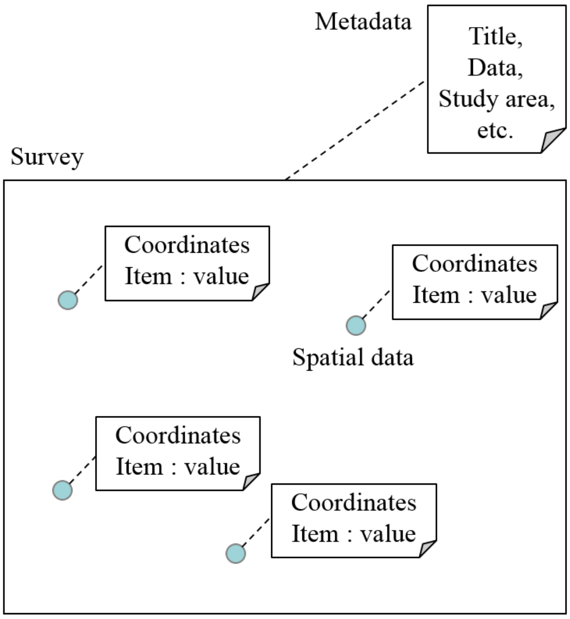


Figure 1. Data structure.

3.3. *Workflow*

Second, we analyze the system requirements based on the workflow of a sensing experiment. It is defined that a sensing experiment consists of three steps: (A) Preparation, (B) Data-acquisition, and (C) Application. This workflow is shown in Figure 2.

step (A) Preparation First, an organizer plans and prepares for the experiment. The organizer decides the operating procedure based on the purpose, including the study area, the survey items, and the schedule. The metadata of the experiment should be stored in the database in this step.

step (B) Data-acquisition Participants of the experiment obtain spatial data in the field, on the basis of the operating procedure decided in step (A). They can focus on the spatial data-acquisition process, and do not need to edit the metadata. As a sensing experiment is usually carried out by multiple fieldworkers at the same time, it is desirable that they can access the system and record/display the obtained data in real-time without an error or overlap.

step (C) Application After the data-acquisition, the obtained data are displayed, analyzed, exported, and used by users for their own purposes. The data users can include the organizer, participants, and others. It is important to select the data sharing policy based on the intention of the organizer. Some data are assumed to be private such as those for commercial use, and others are assumed to be public such as a CC-BY-SA license.

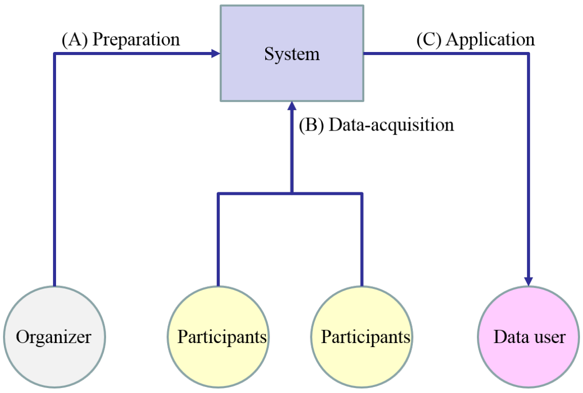


Figure 2. Workflow of the sensing experiment.

The use case of the FPSS should be optimized for this general workflow of a sensing experiment.

3.4. System Architecture

A client–server model is suitable for the system architecture, because multiple fieldworkers are expected to participate in the same sensing experiment at the same time. The client terminals are assumed to be smartphones and tablet PCs equipped with GNSS (Global Navigation Satellite System) including GPS, and other sensors. The fieldworkers travel to various fields with the client terminals, and access the server with wireless LANs or a mobile network such as 3G and LTE (Long Term Evolution). Although the client terminals need to access a remote server, network connectivity to a server is not always guaranteed in an outside field. Therefore, it is desirable for the system to compensate for the lack of network connectivity.

4. Framework for Participatory Sensing Service

The proposed FPSS is described in this section.

4.1. Architecture

We describe the architecture of the proposed framework, which is designed to meet the requirements above. Figure 3 shows the architecture of the proposed framework. It consists of the spatial database, the GIS server, the PSS module, and the client application.

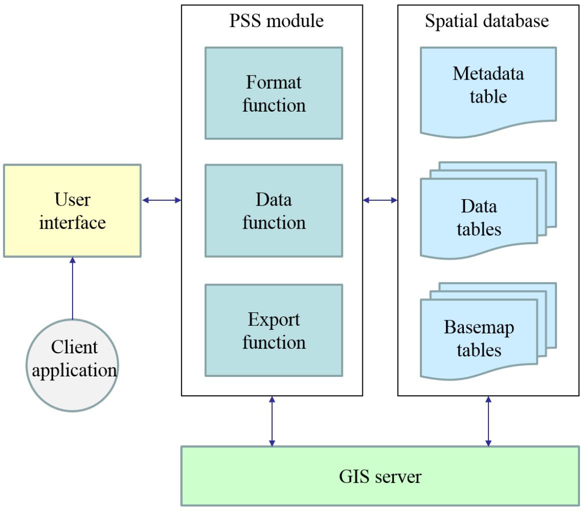


Figure 3. Architecture of the proposed framework.

4.1.1. Spatial Database

The spatial database stores metadata, spatial data, and basemaps. It has a metadata table, data tables, and basemap tables.

Metadata are stored in the metadata table, in which each record represents respective experiments.

Spatial data are stored in the data tables. The spatial data include the coordinates and values of survey items. The proposed framework creates a new data table with optimized fields for each experiment. The created table is linked to the corresponding metadata. With this table structure, the proposed framework supports various types of sensing experiments.

Basemaps are stored in the basemap tables. Basemaps include general open data which are registered in advance, and specific basemap layers of each study area uploaded by system users.

4.1.2. GIS Server

The GIS server provides a GIS engine and protocols for serving georeferenced map images, i.e., Web Map Service (WMS). This works in the back-end of the system and the interface to users is provided by the PSS module.

4.1.3. PSS Module

The PSS module is composed of three components: the format function, the data function, and the export function. They generate user interfaces, store data into the database, select data from the database, and render maps using the GIS server. Each component corresponds to the three steps in the workflow of a sensing experiment described in Section 3.3. The detailed functions are described along with the use case in Section 4.2.

4.1.4. Client Application

The client-side application provides a format-download function and a data-store function, to realize the off-line use of the client terminals.

4.2. Use Case

The sequence of the use case of the proposed framework is shown in Figure 4. The sequence consists of three steps: (1) Format creation; (2) Data input; (3) Export. These steps correspond to the workflow of a sensing experiment in Section 3.3; step (1) corresponds to step (A); step (2) corresponds to step (B); and step (3) corresponds to step (C).

In step (1) users upload and register basemaps to the database and create an experiment format.

In step (2) users download the created format and input the obtained data into the database using the format.

In step (3) users select the format to display as a map and download the selected data.

The novelty of the proposed framework is that it enables fieldworkers of various types of participatory sensing experiments to log data easily in an integrated way. With this architecture and use case, the users are able to create the optimal format for various types of sensing experiments. They can also input and export spatial data in a unified manner. The created format and stored data can be shared with other users. Accordingly, multiple fieldworkers are able to participate in the same experiment simultaneously. Each step is described in detail below.

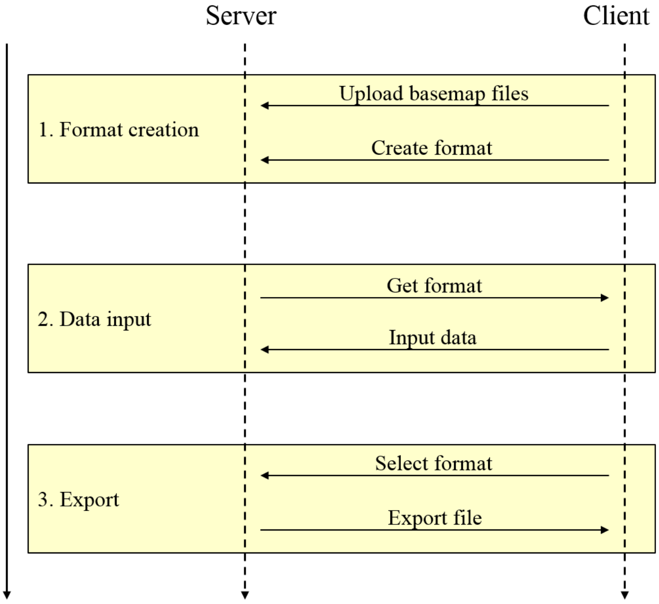


Figure 4. Sequence of the participatory sensing experiment.

4.2.1. Format Creation

The user accesses the server and creates a format for the experiment in the (A) Preparation step. Figure 5 shows the data flow between the system components. When the user starts this step, the format function creates a formatting web form as a user interface. The user fills out the formatting form with the experiment information. The metadata, i.e., the title, date, and notes, are stored in the metadata table. The basemaps are selected from the basemap table and the study area is set using map images. If the user wants to add a new basemap, a new basemap file can be uploaded and stored in the database in advance. The survey items are added designating their name and data types, and a corresponding table is created as a data table. The proposed framework can be employed for various types of sensing experiments because any survey item can be added in this step. Then, the sharing policy of this format is selected.

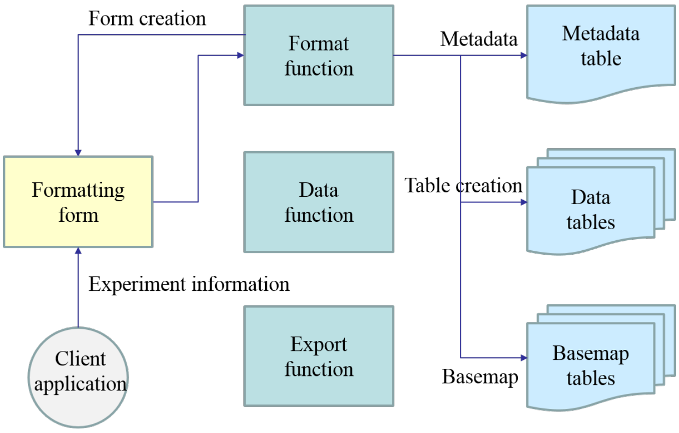


Figure 5. Use case (1): Format creation.

4.2.2. Data Input

In the (B) Data-acquisition step, the user stores the obtained data using the client terminals. Figure 6 shows the data flow in this step. When the user selects a survey from the list of editable surveys, an input form is created by the data function. The data function creates it by reading the metadata from the metadata table. The stored spatial data are read from the data table of this survey and displayed on the basemap images. When the user obtains a datum in the field, the user stores it using the input form. The user plots the location on the map image and fills out the required fields for this data table. An image file can be uploaded if a corresponding web form is created. If the user selects the displayed datum, the stored values are displayed in the form and they can be updated or deleted. Multiple fieldworkers can share the latest result in real-time, because the displayed spatial data are refreshed when the input form is reloaded. If the network connection to the server is unavailable, the obtained data are temporarily stored in the memory of the client terminals. They are automatically uploaded to the server when the network becomes available.

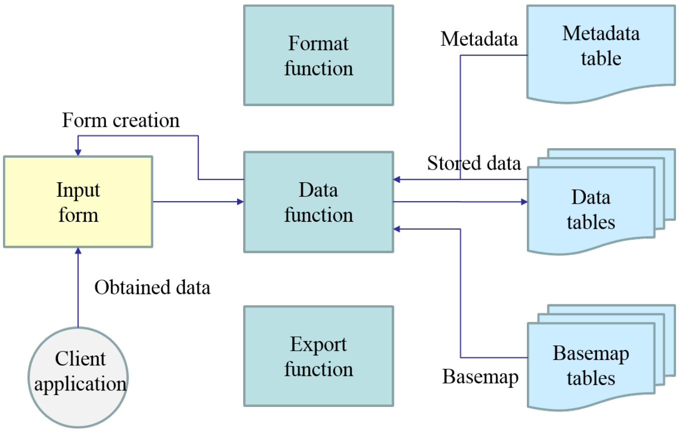


Figure 6. Use case (2): Data input.

4.2.3. Export

In the (C) Application step, the user displays and exports the stored data and uses them for the purpose of the experiment. Figure 7 shows the data flow in this step. When the user selects an experiment from the list of viewable experiments, an export form is created by the export function. The export function creates it by reading the metadata from the metadata table and the stored data from the data tables. The user can view the stored data on the basemap images. When the user requests to download the stored spatial data, the export module translates the spatial data to a file format.

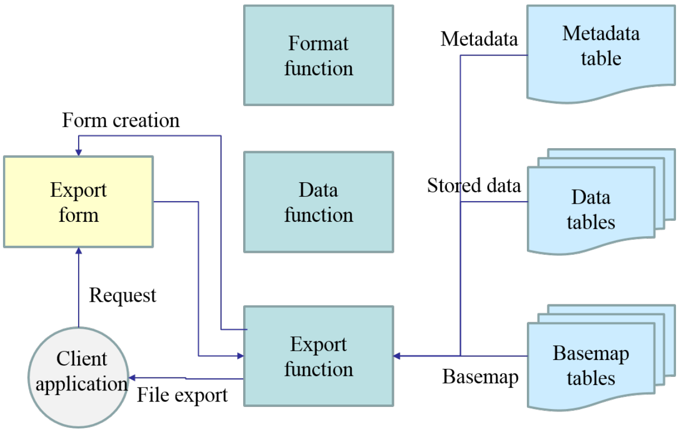


Figure 7. Use case (3): Export.

5. Prototype Implementation

In this section, we describe an open source-based prototype implementation of the proposed framework. As stated before, the purpose of the prototype is to confirm the validity of the proposed framework. Thus, the implementation of the proposed concept is not limited to this; different styles using different tools can be considered.

5.1. Overview

Here, we provide an overview of the implementation. As a prototype of the proposed framework, an open source-based approach is adopted to reduce the system installation cost. We implemented the prototype based on a filed survey assisting system in [13].

Here, we provide an outline of the filed survey assisting system. It is an open source-based web GIS package. It has two features that improve the usability for non-technical users [27]. One is web GIS architecture and the other is that open source GIS applications are wrapped with a content management system (CMS). Web GIS encourages users to employ the system because they can do so simply with web browsers, with which they are familiar [28]. In general, a CMS makes it easy for users to employ the system. The CMS manages web content in an integrated fashion and conceals complicated GIS functions from users. Users work with interfaces formed by the CMS, without having to struggle with or even be conscious of open source GIS applications.

5.2. Architecture

The architecture of the implemented system is presented in Figure 8.

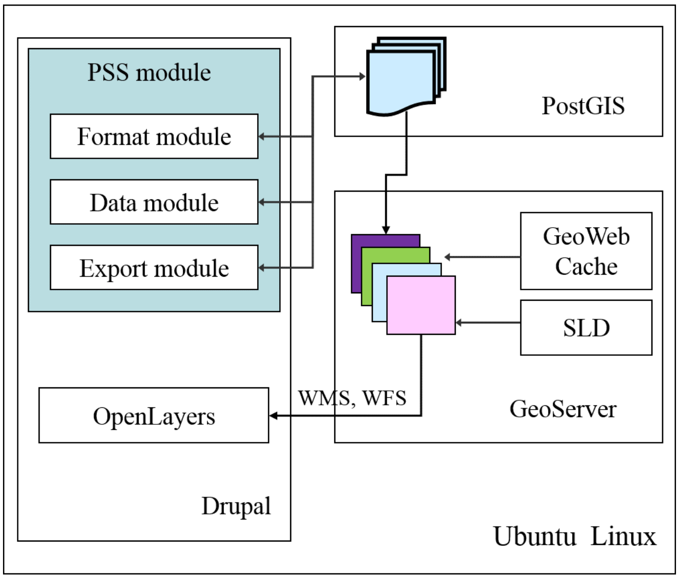


Figure 8. Architecture of the implemented system.

In the server, open source GIS applications and Drupal (CMS) run on Ubuntu Linux. Spatial data are stored using PostGIS. PostGIS is a spatial extension of PostgreSQL, which is one of the most popular open source database management systems. GeoServer works as a GIS server, which is a major web GIS application built on open standards and enables the interoperability of spatial data. It supports WMS, which delivers maps as images, and Web Feature Service (WFS), which delivers maps in Geography Markup Language (GML). These data formats are standardized by Open Geospatial Consortium (OGC) and used in many GIS applications. A Styled Layer Descriptor (SLD) describes the appearance of the map layers. GeoWebCache accelerates processing by caching requests. Drupal works in an integrated way as a user interface and manages GIS applications. We employed Drupal because it provides spatial modules including OpenLayers, which displays and renders maps on web pages using WMS and WFS.

The PSS module is developed as a module of Drupal and is written in PHP scripts. It creates user interfaces and controls database queries. The PSS module has three sub-modules: a format module, a data module, and an export module. The format module corresponds to the use case (1) Format creation; the data module corresponds to (2) Data input; and the export module corresponds to (3) Export.

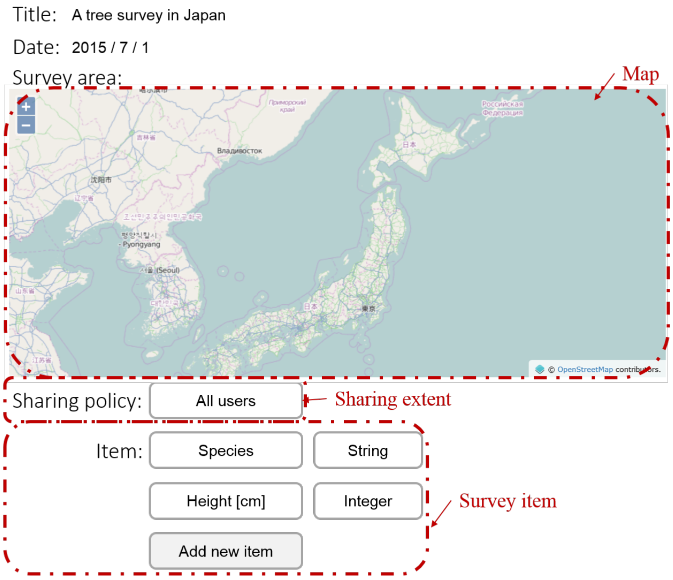
The users access the server with their client terminals. They can view the downloaded formats using a client application, which provides the data-store function. The location of the user is displayed in the map images using GPS. The transmission protocol used between the server and clients is HTTP(S).

5.3. Use Case

Here, we describe the detail of the PSS module along with the use case. The PSS module is developed and installed as a Drupal module.

5.3.1. Format Creation

Figure 9 shows an example screenshot of the formatting form, which is created by the format module. The basic web functions such as account management and database queries are developed using the Drupal functions. The sharing policy of a created format is also easily controlled with the Drupal functions. The map images are provided with WMS using GeoServer and are displayed with OpenLayers. The basemaps are selected from Google Maps (Google Inc., Mountain View, CA, USA), VirtualEarth (Microsoft Corporation, Redmond, WA, USA), and OpenStreetMap (OSM, [29]). The users can also upload Shapefiles and GeoTIFFs as basemaps. The uploaded files are imported into the database using a shp2pgsql or a raster2pgsql tool.



Title: A tree survey in Japan

Date: 2015 / 7 / 1

Survey area:

Sharing policy: All users

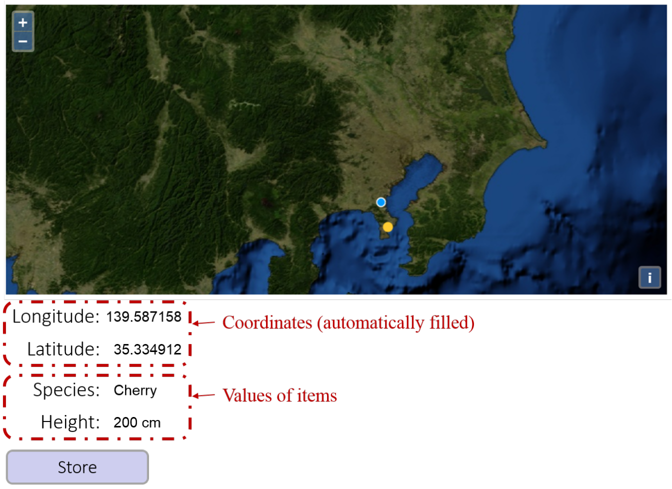
Item: Species String

Height [cm] Integer

Figure 9. Example screenshot of format creation.

5.3.2. Data Input

When the user logs in to the system and requests the editable experiment list, the list of editable experiments is generated using the account and content management function of Drupal. Figure 10 shows an example screenshot of the input form, which is created by the data module. The data module calls queries to the corresponding tables in the database, and displays the map images with GeoServer and OpenLayers. When the user plots a new datum on the map, the system automatically obtains the coordinates and sends a query to the database to store it. The location is automatically plotted in the map, if the GPS function is available.



Longitude: 139.587158
Latitude: 35.334912

Species: Cherry
Height: 200 cm

Coordinates (automatically filled)

Values of items

Figure 10. Example screenshot of data input.

If the network connection between the server and the client terminal becomes unavailable, the client application stops the request to the data module and continues to display the last information it downloaded from the server. Until the network connection becomes available, new data are stored in the client terminal's own memory. Then, the stored data are automatically uploaded to the server.

5.3.3. Export

When the user requests the viewable experiment list, it is generated with the CMS functions based on the user's account information. The export module calls queries to the database and displays the map images with GeoServer and OpenLayers. If the user wants to export the stored data, either CSV (Comma-Separated Values) or KML (Keyhole Markup Language) can be selected as the export file type. The export module translates the spatial data to the file using the PostGIS functions.

6. Evaluation

6.1. Outline

We evaluated the implemented framework through a test experiment. We confirmed that the participatory sensing experiment was performed smoothly by multiple fieldworkers with the proposed framework.

6.1.1. Study Area and Item

As an application study of the proposed framework, we undertook a growth survey of Kawazu-zakura tree (Prunus lannesiana Wils.) in Miura Peninsula, Kanagawa, Japan. The study area, shown in Figure 11, covers an area of about 500 m^2 from Keikyu-Miurakaigan Station to Komatsugaike Park. The location and height of the Kawazu-zakura trees were adopted as survey items.

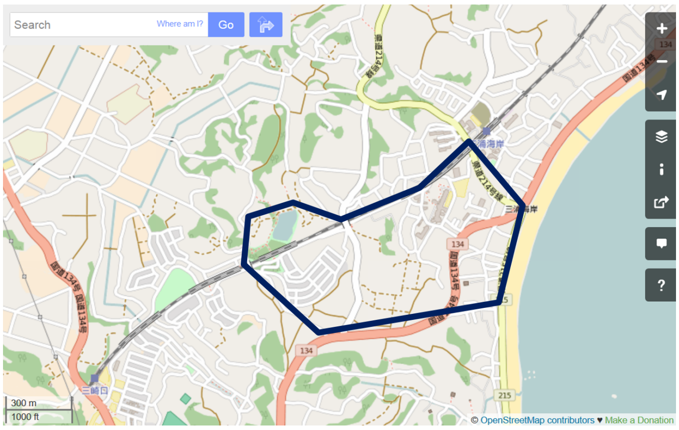


Figure 11. Study area (on OpenStreetMap).

6.1.2. Server

We located the server in a remote building. We stored the IP address in the server and connected it to the Internet. The client terminals accessed the server via the Internet. OSM tiles were installed in the server and used as the basemap.

6.1.3. Client

There were ten participants in the survey. They were divided into four groups. Each group member had an Android 4.2 based smartphone (LTE network), an Android 4.2 based tablet (3G network), an iPhone 5 (LTE network), and a laptop PC (WiMAX network). At first, the Android terminals were used without network services. For the latter half of the survey, they were used with the network services. Other client terminals were always used with Internet access. With this configuration, the data-store function of the client application was verified.

6.1.4. Procedure

One of the participants created an experiment format in the server in advance. The created format was shared among all system users. For the Android terminal users, the created format was downloaded to their terminals. The participants accessed the server with the client applications and logged in using their own accounts.

The participants walked around the study area and recorded the location and height of Kawazu-zakura trees based on visual observation. When the participants found a Kawazu-zakura tree, they opened the created format and input the obtained data. When the participants plotted a point on the map, the coordinate forms were automatically supplied with the point's location.

After the experiment, we employed a questionnaire to elicit users' impressions of the system. The question items related to the overall impression of the experiment, the advantages of the proposed system, and the points to be improved. They were open-ended questions.

6.2. Results

The experiment was completed smoothly. About 500 Kawazu-zakura trees were plotted in the study area within four hours. The participants were able to use the system simultaneously with their client terminals. The transmission and processing delays of the system were slight. Although sometimes it was difficult for some participants to distinguish tree species, they encountered no problems when operating the developed system, because they had only to plot points and input tree heights. The stored data were confirmed with a WMS request and exported as a KML file.

With the network access to the server, the participants were able to access the results in real-time. They could see the data that they had stored and also the data that other participants had stored.

The experiment results were updated automatically. However, while the network access to the server was unavailable, the input form cannot be updated with newly stored data. Thus, once all the participants activated the network services and the locally stored data were uploaded, it emerged that some data overlapped. This is because a participant plotted certain trees already plotted by another participant.

Figure 12 shows the relationship between the number of data and the data size stored in the system. The stored data consist of the point ID (4 bytes), coordinates (16 bytes), and height (4 bytes) per point. The data size increases in accordance with the number of data points. In this case study, 500 points data occupy 10 kilobytes in the system.

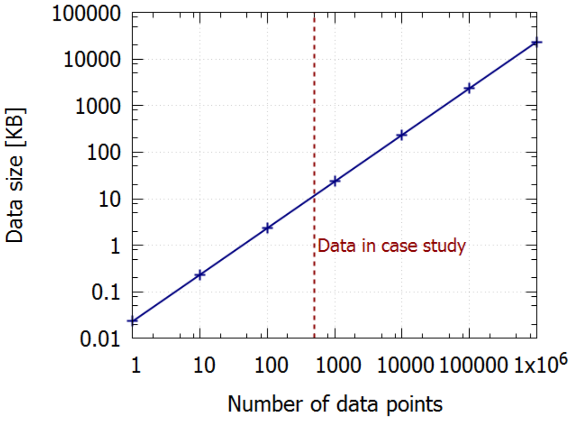


Figure 12. Data size stored in the system.

6.3. Discussion

Here, we discuss the advantages and limitations of the proposed FPSS using the results of the questionnaire.

The results of this application study suggest that a sensing experiment can be performed efficiently with the proposed framework. The data structure and use case of the framework were suitable for participatory sensing services. Multiple fieldworkers could use the same format that was created specifically for the experiment, using various types of client terminals. The system simplified the operation procedure of the sensing and it was easy to use. Because the progress of the experiment was updated in real-time, participants could see how other participants were proceeding with the experiment, and compare their results with those obtained by other participants.

The open source GIS-based implementation was sufficient for the test experiment. The incorporation of web GIS architecture and CMS improved the usability, and the implemented system was easy to use for all participants. If the data size and work load of the system become very large, the system architecture should be reconsidered.

The limitation of the FPSS was the data overlap; some points that a participant plotted without the network service overlapped points plotted by others. This overlap problem is inevitable with this type of system. This is because the obtained data are not synchronized with those of other participants until the network service becomes available. This problem can be prevented by putting a mark on the trees that have already been plotted on the map.

7. Conclusions

This paper introduced a web GIS framework for participatory sensing service based on the comprehensive analysis of the requirements. The proposed FPSS enables an integrated deployment of spatial data capture, storage, and data management functions. In various types of participatory sensing experiments, users can collect and manage spatial data in a unified manner. It can be used across multiple scales; an individual, a group, or a community.

This feature is realized by the optimized system architecture and use case based on the general requirements for participatory sensing. We defined that participatory sensing consists of three steps: (A) Preparation; (B) Data-acquisition; and (C) Application. The sequence of the use case of the proposed framework consists of three steps, namely, (1) Format creation; (2) Data input; and (3) Export, and they correspond to the three steps of participatory sensing. The architecture of the proposed framework consists of the spatial database, the GIS server, the PSS module, and the client application. Then, we developed an open source GIS-based implementation of the proposed framework, which can overcome financial difficulties which are one of the major problems of deploying sensing experiments.

With the prototype, we confirmed that participatory sensing experiments can be performed efficiently with the proposed FPSS. Multiple fieldworkers could easily use the same format that was created specifically for the experiment. They could share the state of the sensing in real-time. If the network service was unavailable, the obtained data were stored in their client terminals and uploaded immediately once the network service became available.

We believe that this work will contribute to the progress and deployment of GIS technologies for participatory sensing. A quantitative evaluation of usability is a future task.

Acknowledgments: This work was supported by JSPS KAKENHI Grant Number JP15H02722.

Author Contributions: Yu Nakayama organized the proposed concepts and wrote the paper; Kazuhiko Nakamura, Hitoshi Saito, and Rui Fukumoto implemented the prototype and performed the experiments

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lane, N.D.; Miluzzo, E.; Lu, H.; Peebles, D.; Choudhury, T.; Campbell, A.T. A survey of mobile phone sensing. *IEEE Commun. Mag.* **2010**, *48*, 140–150.
2. Khan, W.Z.; Xiang, Y.; Aalsalem, M.Y.; Arshad, Q. Mobile phone sensing systems: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 402–427.
3. Wang, L.; Zhang, D.; Wang, Y.; Chen, C.; Han, X.; M'hamed, A. Sparse mobile crowdsensing: Challenges and opportunities. *IEEE Commun. Mag.* **2016**, *54*, 161–167.
4. Ma, H.; Zhao, D.; Yuan, P. Opportunities in mobile crowd sensing. *IEEE Commun. Mag.* **2014**, *52*, 29–35.
5. Burke, J.A.; Estrin, D.; Hansen, M.; Parker, A.; Ramanathan, N.; Reddy, S.; Srivastava, M.B. Participatory sensing. *Center Embed. Netw. Sens.* **2006**. Available online: http://escholarship.org/uc/item/19h777qd#page-5 (accessed on 8 April 2017).
6. Estrin, D. Participatory sensing: Applications and architecture [internet predictions]. *IEEE Int. Comput.* **2010**, *14*, 12–42.
7. Sheng, X.; Xiao, X.; Tang, J.; Xue, G. Sensing as a service: A cloud computing system for mobile phone sensing. In Proceedings of the 2012 IEEE Sensors, Taipei, Taiwan, 28–31 October 2012; pp. 1–4.
8. Brouwers, N.; Langendoen, K. Pogo, a middleware for mobile phone sensing. In Proceedings of the 13th International Middleware Conference, Montreal, QC, Canada, 03–07 December 2012; Springer: New York, NY, USA, 2012; pp. 21–40.
9. Zavala-Romero, O.; Ahmed, A.; Chassignet, E.P.; Zavala-Hidalgo, J.; Eguiarte, A.F.; Meyer-Baese, A. An open source Java web application to build self-contained web GIS sites. *Environ. Model. Softw.* **2014**, *62*, 210–220.
10. Agrawal, S.; Gupta, R.D. Development and Comparison of Open Source based Web GIS Frameworks on WAMP and Apache Tomcat Web Servers. *Proc. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *4*, 1–5.
11. Tiwari, A.; Jain, D.K. Geospatial Framework For Dengue using Open Source Web GIS Technology. In Proceedings of the Joint International Workshop of ISPRS WG VIII/1 and WG IV/4 on Geospatial Data for Disaster and Risk Reduction, Hyderabad, India, 21–22 November 2013; pp. 21–22.
12. Chakraborty, D.; Sarkar, D.; Agarwal, S.; Dutta, D.; Sharma, J.R. Web Based GIS Application using Open Source Software for Sharing Geospatial Data. *Int. J. Adv. Remote Sens. GIS* **2015**, *4*, 1224–1228.
13. Nakayama, Y.; Mori, S. FOSS4G Based Mobile Web-GIS for Field Survey in Natural Environmental Studies. In Proceedings of the 9th International Conference on Ubiquitous Intelligence & Computing and 9th

International Conference on Autonomic & Trusted Computing (UIC/ATC), Fukuoka, Japan, 4–7 September 2012; pp. 707–712.

14. Reddy, S.; Burke, J.; Estrin, D.; Hansen, M.; Srivastava, M. A framework for data quality and feedback in participatory sensing. In Proceedings of the 5th International Conference on Embedded Networked Sensor Systems, Sydney, Australia, 4–9 November 2007; ACM: New York, NY, USA, 2007; pp. 417–418.
15. Guo, B.; Yu, Z.; Zhou, X.; Zhang, D. From participatory sensing to mobile crowd sensing. In Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Budapest, Hungary, 24–28 March 2014; pp. 593–598.
16. Fakoor, R.; Raj, M.; Nazi, A.; Di Francesco, M.; Das, S.K. An integrated cloud-based framework for mobile phone sensing. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 17 August 2012; pp. 47–52.
17. Kazemi, L.; Shahabi, C. A privacy-aware framework for participatory sensing. *ACM SIGKDD Explor. Newsl.* **2011**, *13*, 43–51.
18. Christin, D.; Roßkopf, C.; Hollick, M.; Martucci, L.A.; Kanhere, S.S. Incognisense: An anonymity-preserving reputation framework for participatory sensing applications. *Pervasive Mob. Comput.* **2013**, *9*, 353–371.
19. Cardone, G.; Foschini, L.; Bellavista, P.; Corradi, A.; Borcea, C.; Talasila, M.; Curtmola, R. Fostering participaction in smart cities: A geo-social crowdsensing platform. *IEEE Commun. Mag.* **2013**, *51*, 112–119.
20. Cardone, G.; Cirri, A.; Corradi, A.; Foschini, L. The participact mobile crowd sensing living lab: The testbed for smart cities. *IEEE Commun. Mag.* **2014**, *52*, 78–85.
21. Cardone, G.; Cirri, A.; Corradi, A.; Foschini, L.; Maio, D. MSF: An efficient mobile phone sensing framework. *Int. J. Distrib. Sens. Netw.* **2013**. Available online: http://journals.sagepub.com/doi/full/10.1155/2013/538937 (accessed on 8 April 2017).
22. Guo, B.; Wang, Z.; Yu, Z.; Wang, Y.; Yen, N.Y.; Huang, R.; Zhou, X. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Comput. Surv. (CSUR)* **2015**, *48*, 7.
23. Pournajaf, L.; Xiong, L.; Sunderam, V.; Goryczka, S. Spatial task assignment for crowd sensing with cloaked locations. In Proceedings of the IEEE 15th International Conference on Mobile Data Management (MDM), Brisbane, Australia, 15–18 July 2014; Volume 1, pp. 73–82.
24. Foremski, P.; Gorawski, M.; Grochla, K.; Polys, K. Energy-efficient crowdsensing of human mobility and signal levels in cellular networks. *Sensors* **2015**, *15*, 22060–22088.
25. Liu, Y.; Niu, J.; Liu, X. Comprehensive tempo-spatial data collection in crowd sensing using a heterogeneous sensing vehicle selection method. *Pers. Ubiquitous Comput.* **2016**, *20*, 397–411.
26. Zavala-Romero, O.; Chassignet, E.P.; Zavala-Hidalgo, J.; Velissariou, P.; Pandav, H.; Meyer-Baese, A. OWGIS 2.0: Open source Java application that builds web GIS interfaces for desktop and mobile devices. In Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Dallas/Fort Worth, TX, USA, 4–7 November 2014; ACM: New York, NY, USA, 2014; pp. 311–320.
27. Nakayama, Y.; Nakamura, K.; Saito, H.; Fukumoto, R. Open cafe system: Development and application of FOSS4G package in natural environmental field. *Theory Appl. GIS* **2011**, *19*, 17–23. (In Japanese with English Abstract)
28. Iwasaki, N.; Sprague, D.S.; Koyanagi, T.; Furuhashi, T.; Yamamoto, S. Development of the historical agro-environment browsing system constructed by FOSS4G. *Theory Appl. GIS* **2009**, *17*, 83–92. (In Japanese with English Abstract)
29. OpenStreetMap. Available online: http://www.openstreetmap.org/ (accessed on 1 Feburuary 2017).