

4W1H in Mobile Crowd Sensing

Daqing Zhang, Leye Wang, Haoyi Xiong, and Bin Guo

ABSTRACT

With the rapid proliferation of sensor-rich smartphones, mobile crowd sensing has become a popular research field. In this article, we propose a four-stage life cycle (i.e., *task creation*, *task assignment*, *individual task execution*, and *crowd data integration*) to characterize the mobile crowd sensing process, and use 4W1H (i.e., *what*, *when*, *where*, *who*, and *how*) to sort out the research problems in the mobile crowd sensing domain. Furthermore, we attempt to foresee some new research directions in future mobile crowd sensing research.

INTRODUCTION

Mobile crowd sensing (MCS), a term coined by Ganti *et al.* [1], has recently spurred lots of research interest. Similar to the notion of participatory sensing and human-centric computing [2], MCS refers to the *sensing paradigm* in which mobile users with sensing and computing devices are tasked to collect and contribute data in order to enable various applications. MCS applications leverage the sensing, computing, and wireless communication capability offered by the millions of mobile devices (e.g., Android phones, iPhones, and iPads) already deployed in the fields and carried by people in their daily lives. MCS has successfully extended the sensing scope from a single physical space to the community- and city-wide scales, from recognizing hazardous environmental situations to gaining knowledge on the collective behavior of crowds.

Thus far, two extensively studied subareas in MCS research are MCS framework and relevant applications. While a lot of effort has been devoted to creating various real-world crowd sensing applications, such as semantic place characterization [3], environmental influence monitoring [4], and social interaction sensing [5], several MCS frameworks handling different design issues such as participatory management [6], programming interface [7], and user privacy have been developed. Among those frameworks, MEDUSA [7] is probably the most cited, which focuses on facilitating MCS *task/application creation*, *task assignment*, and *sensing data collection*.

Despite the great number of works in MCS literature, a systematic study and classification of the research problems in the MCS research domain is lacking to guide further research and development of this emerging field. To this end, we attempt to fill the gap by dividing the life

cycle of the MCS process into four stages and using **4W1H** (i.e. *what/when/where/who/how*) to characterize the major research issues in each of the four stages of the MCS life cycle as well as across the whole MCS process. More specifically, we intend to make the following contributions in this article:

- Proposing to divide the life cycle of MCS process into four stages: *task creation*, *task assignment*, *individual task execution*, and *crowd data integration* inspired by existing MCS frameworks.

- Proposing the use of 4W1H to characterize the research issues of MCS in a structured way. More specifically, we apply 4W1H in each stage of the MCS life cycle as well as throughout the whole MCS process to identify the research problems locally and globally, providing guidelines and insights for further MCS research.

- Identifying research gaps in future MCS research. By systematically analyzing the existing MCS research work and applications using the four-stage MCS life cycle and 4W1H framework, we summarize the main research efforts in MCS until now and foresee the missing MCS efforts for further research. In particular, we identify several meaningful yet unaddressed research problems by applying the methodology proposed in this study.

MOBILE CROWD SENSING: LIFE CYCLE AND DESIGN DIMENSIONS

LIFE CYCLE OF MOBILE CROWD SENSING

In general, the life cycle of an MCS process consists of *creating MCS applications* according to the requirements, *assigning sensing tasks* to participants, *executing the task* (sensing, computing, and uploading) on the mobile device of an individual participant, and *collecting and processing sensed results* from participants. Inspired by the three-step process (i.e., recruit, execute, and collect) of the MCS framework proposed in Madusa [7], we further divide the life cycle of the MCS process into four phases: *task creation*, *task assignment*, *individual task execution*, and *crowd data integration*, as shown in Fig. 1. The key functionalities of each phase are described below.

Task creation: The MCS organizer creates an MCS task through providing the participants with the corresponding mobile sensing applications to be deployed in the participants' smartphones later. Existing work in the *task creation* stage focuses on simplifying both the MCS task and the corresponding mobile application cre-

Daqing Zhang, Leye Wang, and Haoyi Xiong are with TELECOM Sud-Paris.

Bin Guo is with North-west Polytechnic University, China.

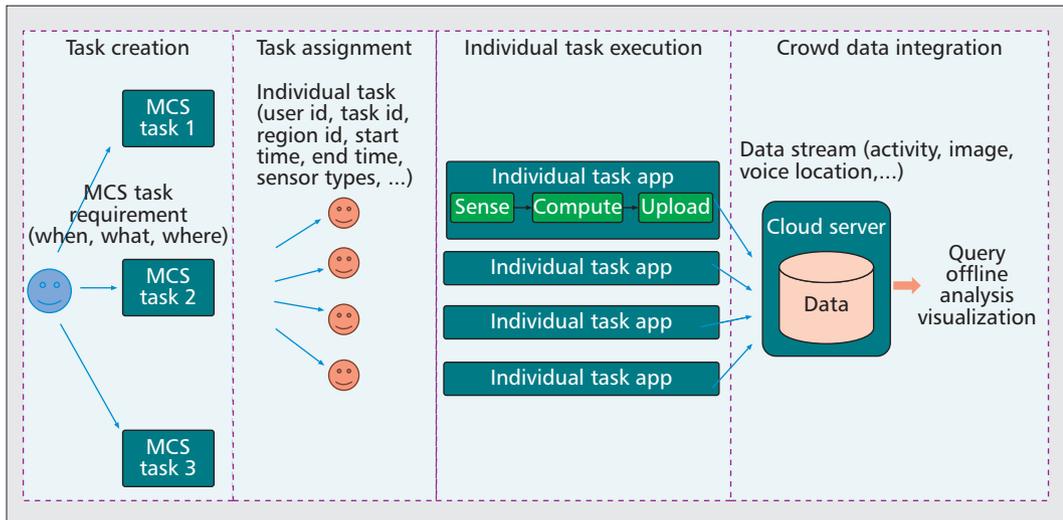


Figure 1. The four-stage life cycle of the mobile crowd sensing process.

ation process for MCS organizers, usually by offering an easily understood domain-specific language (DSL), such as MedScript, which is an XML-based programming language in the MEDUSA framework [7].

Task assignment: After the organizer creates an MCS task and the corresponding mobile task applications, the next phase is *task assignment* — recruiting participants and assigning them the individual sensing tasks that are supposed to run on each participant’s mobile device. Finding enough and appropriate crowd sensing participants is the core issue in this stage. A widely used method to recruit crowd sensing participants is leveraging a traditional crowdsourcing platform, such as *Amazon Mechanical Turk*. Existing research work also proposes various schemes to select a subset of participants in order to achieve different optimization goals, such as ensuring area coverage, minimizing task completion time and number of participants [8], and choosing high-reputation/performance participants [6].

Individual task execution: Once receiving the assigned sensing task, a participant will try to finish it within a predefined MCS task duration in parallel with other tasks. This phase is called the *individual task execution* stage, which can be further divided into three sub-stages: *sensing*, *computing*, and *data uploading*. Although individual tasks might vary greatly depending on the specific MCS applications, existing work tries to offer high-level sandboxed running environments to reduce smartphone energy consumption or speed up individual task execution through mechanisms such as piggybacking [9] and computation/code offloading [5].

Crowd data integration: This stage takes the data streams collected from all the participants as input, aggregates the data, and provides end users with what they need in the appropriate format. For some MCS applications [10], the data processing in this stage is quite straightforward — a central server stores the data and provides interface to end users for data query and sharing. While other MCS applications [3–5] employ complicated algorithms to integrate data,

pivot data, and extract high-level and collective intelligence from the raw data of large crowds.

4W1H IN MOBILE CROWD SENSING

In order to explore all the MCS research dimensions, in this article we propose to use 4W1H to describe the research objectives, issues, and solutions in a structured manner. For a real-time highway traffic congestion monitoring MCS application in rush hours, we could use the following 4W1H to specify the MCS application:

- *What*: traffic congestion status
- *When*: 07:00–09:00 and 17:00–19:00
- *Where*: a specific road segment in a certain highway
- *Who*: volunteering vehicular drivers
- *How*: real-time and location-based vehicle speed collection, aggregation, and traffic status updating

Taking the proposed four-stage life cycle of the MCS process into consideration, the 4W1H framework can actually be applied to each of the four stages. Take the aforementioned highway traffic congestion monitoring MCS application as an example:

- The first stage focuses on facilitating the MCS task creation based on specification and requirements of highway traffic congestion detection. Obviously, three main factors should be considered: *what* (i.e., traffic congestion status) should be sensed; *where* and *when* (i.e., on a highway and 07:00–09:00/17:00–19:00, respectively) the MCS task should be executed.

- Suppose the MCS task has been successfully created based on the given specification. Generally, a number of MCS participants will be selected to receive the task assignments. Thus, the major issue here is *who* (i.e., volunteering drivers) should receive the sensing tasks.

- After participants receive the sensing tasks, they execute the tasks. Therefore, the major issue here is *how* sensing tasks are implemented and executed by each participant’s device. Specifically, issues such as *what to sense*, *how frequent the sensing should be*, *how participants should collaborate to reduce sensing, computing and uploading efforts*, and so on are handled at this stage.

In general, the life cycle of a mobile crowd sensing application consists of creating MCS applications according to the requirements, assigning sensing tasks to participants, executing the task on the mobile devices of individual participants, and collecting and processing sensed results from participants.

In the task creation stage, the MCS organizer defines the MCS task and designs/implements the individual mobile sensing application. The key issue in task creation is to specify the objective (what), timing (when), and coverage area (where) for a specific MCS task.

•After the execution of each individual's task, the crowd data integration process starts. The main issue here is *how* we integrate the collected sensed results (e.g., speed/location/timestamp of individual participants) and disseminate the final results (i.e., congestion status) to end users.

It is apparent from the above example that the 4W1H framework not only applies to the overall MCS design, but also to each stage of the MCS life cycle. In the next section, we elaborate on the 4W1H issues at each stage of the MCS life cycle in detail.

4W1H IN EACH STAGE OF THE MCS LIFE CYCLE

4W1H IN TASK CREATION

In the *task creation* stage, the MCS organizer defines the MCS task and designs/implements the individual mobile sensing application. The key issue in task creation is to specify the objective (*what*), timing (*when*), and coverage area (*where*) for a specific MCS task. Most of the research work on task creation focuses on helping an MCS organizer without programming skill set to create MCS tasks by simplifying the task creation process. Correspondingly, the 4W1H issues in the task creation stage include:

What: This refers to what applications the MCS tasks intend to support; it roughly contains two broad categories:

•**Urban computing:** Many MCS tasks address urban-scale problems such as environment monitoring, traffic condition monitoring, and location characterization. For example, PEIR [4] generates a personal environment influence report for each participant by collecting one's GPS data and other context data (e.g., weather and traffic). CrowdSense@Place [3] links a location to place categories (e.g. store, restaurant) by aggregating opportunistically captured images and audio clips from participants' smartphones.

•**Social computing:** The other category of MCS tasks aim to reveal society-scale issues by collecting participants' daily life traces and social interactions. In SociableSense [5], for instance, users are provided with a quantitative measure of their sociability via their sensed office behavior from smartphones.

Where/when: Where and when refer to the *spatial* and *temporal* features of MCS tasks, they can be classified into the following four categories:

•**Short-range short-term:** This category corresponds to the MCS task executed by participants in physical proximity for a short time, for example, sensing activities and face-to-face interactions among participants in a conference.

•**Long-range short-term:** It corresponds to the MCS task executed by participants staying far apart for a short time (e.g., collecting users' captured images and audio clips across the city for rescue when flooding occurs).

•**Short-range long-term:** This corresponds to the MCS task executed by participants in physical proximity for a long time, for example, sensing students' activities and interactions in a school for several weeks or semesters.

•**Long-range long-term:** This category corresponds to the MCS task executed by participants staying far apart for a long time, such as when engaging citizens to monitor the environmental conditions (e.g., air quality, noise) of a city for several months.

Who: Currently, MCS task organizers are usually large organizations, such as academic institutions, government agencies, and business corporations, as creating the MCS task and platform is quite tedious. When appropriate tools and platforms for task creation are ready, individual users are expected to create an MCS task easily.

How: The most common research issue in task creation is *to simplify the task creation process for the MCS organizer* so that more people with little or no programming skill could also create MCS tasks. To facilitate the task creation process, MEDUSA [7] provides an xml-based domain-specific language to ease the application development, while McSense [8] offers a GUI to create MCS tasks.

4W1H IN TASK ASSIGNMENT

In the *task assignment* stage, the MCS organizer recruits and selects participants for the MCS task. The 4W1H issues in task assignment stage are:

Who: The fundamental problem in task assignment is to find enough and appropriate participants to fulfill the MCS task ("*Who should be selected for participating MCS tasks?*"), i.e., the participant recruitment and selection mechanism. Participants are usually recruited on a volunteering basis with certain incentives offered, often some of them are selected based on criteria which can optimize certain objective functions (e.g. *maximizing area coverage* or *selecting high-reputation persons*).

What: "What" refers to what type of platforms the MCS organizer uses to recruit participants. There are generally two types of recruitment platforms:

•Open public platforms, where MCS organizers publish the tasks and incentives to the public on-line, and the users decide if they want to participate in a certain task according to their price and availability [7]. For example, *Amazon Mechanical Turk* is a popular platform for user recruitment.

•Mission-oriented close platforms, where candidate participants agree to be selected for certain MCS tasks under agreed terms and conditions, the MCS organizer selects an optimal subset of the participants to actually conduct the MCS tasks [6, 8], leveraging candidate participants' personal profiles and historic information (e.g. mobility patterns, mobile application usage logs). Recent years have witnessed an increasing research interest in choosing the best set of participants. Thus, the "Who" and "What" problems are often interleaved and associated with each other.

Where: "Where" refers to the coverage area of the selected participants. For many location-centric MCS tasks, area coverage is a critical concern. To ensure/maximize the area coverage for an MCS task, recruiting participants needs to consider the users' mobility patterns, so as to

Task assignment					
	Who	Where	When	What (user profile)	How
McSense		Ensure full coverage	Static	Mobility traces	To reduce task completion time/user budget
Participatory Recruitment Framework	Propose specific task assignment schemes, that is, <i>selecting who participates from several candidates</i>	Coverage-based recruitment scheme	Static	Mobility traces, reputation, cost	To maximize coverage (coverage-based); select high reputation participants (reputation-based)
EEMC		Ensure full coverage	Dynamic	Mobility traces, voice call logs	To save total participants' energy consumption
CrowdSense @Place	Study coverage and scalability of place-centric crowdsensing: <i>relationship between user population and coverage</i>		—	Mobility traces	—

Table 1. Some research work in task assignment.

achieve full coverage or high coverage ratio of the target area.

When: This refers to the timing of assigning tasks. Traditional MCS task assignment methods recruit participants before the MCS task starts, which we term *static assignment*. However, some advanced MCS task assignment schemes use *dynamic assignment*, that is, assigning tasks to participants during the MCS task execution process, so the organizer can choose appropriate participants according to the ongoing MCS task execution/completion status for purposes such as saving energy consumption, reducing participant budget, and ensuring area coverage.

How: The research issues in this stage investigate *how to save participants' total energy consumption/reduce task completion time/maximize area coverage/minimize participants' incentives* by using intelligent task assignment methods.

Some existing work on task assignment is listed in Table 1. The recruitment framework in participatory sensing [6] proposes two task assignment schemes, coverage-based and reputation-based, in order to maximize area coverage (*where*) and select high reputation participants (*who*), respectively. To minimize task completion time and participant budget while ensuring area coverage (*where*), McSense [8] examines three assignment schemes (random, attendance, and recency). Unlike these *static assignment* schemes, EEMC [11] proposes a dynamic assignment scheme (*when*) that piggybacks data transmission over voice calls to reduce participants' energy consumption. In EEMC, whether a user should be selected as a participant or not is determined when the user places a phone call, based on the current MCS task execution status and the user's future mobility and phone call predictions. Note that all these works design the MCS recruitment platform (*what*) assuming that users' logs and profiles (e.g., mobility, voice call, and reputation) are available. In order to encourage users' participation in MCS tasks, an MCS organizer usually needs to provide participants with some incentives (e.g., money or mobile communication package). So far, two types of incentive mecha-

nisms have been studied: *organizer pricing* (the organizer decides how much each participant is paid) and *user bidding* (each user bids for a certain MCS task). With respect to the *total incentives* paid to all participants, *the total budget* for the MCS tasks, and *sensing coverage/utility*, game theory has been applied to analyze both incentive mechanisms [12] in order to:

- *Minimize the total incentives* while ensuring the *sensing coverage*
- *Maximize the MCS system's utility* under the *budget constraint*.

In addition to the above works, [3] uses sensed data from 85 participants who deployed the CrowdSense@Place application in mobile phones to study high-level coverage and scalability characteristics of place-centric crowd sensing, especially the relationship between participant population and area coverage in Seoul. The results obtained are helpful in answering questions in task assignment, such as "*how many participants are needed to achieve a certain area coverage?*".

4W1H IN INDIVIDUAL TASK EXECUTION

Individual task execution is the third stage of the MCS life cycle. In this stage, participants conduct the sensing tasks and upload the sensed results to the central server. This stage can be further divided into three sub-stages: *sensing*, *computing*, and *uploading*. The 4W1H issues in this stage are:

What: In this stage, what refers to the objective of the individual task (i.e., *what to sense*, *what to compute*, and *what to upload*). While *what to sense* depends on the MCS application and availability of sensors, *what to compute* relies on how much processing is put on the mobile device and how much on the server; *what to upload* is closely related to *what to sense* and *what to compute*, and is also related to other mobile devices when collaborative sensing is applied.

When: Here, this refers to the timing of conducting a individual task (i.e. *when to sense*, *when to compute*, and *when to upload*). Choosing the

	Sense				Compute				Upload				
	What	When	Where	Who	What	When	Where	Who	What	When	Where	Who	How
Piggyback		X				X				X			Save energy
SociableSense		X					X		X				Save energy
Collaborative crowdsensing			X										Save energy/ ensure coverage
Data compression					X				X				Save energy/ save data cost
effSense										X	X	X	Save energy/ save data cost

Table 2. Some research work on individual task execution.

right timing to do an individual task might help participants obtain high-quality sensed data, reducing computing time, and saving battery life.

Where: For the *sensing, computing, and uploading* sub-stages, where has slightly different meanings. For example, *where to sense* is about the geographical area where sensors need to be triggered; *where to compute* is about where computation is done to process the sensed data (in a smartphone or server); and *where to upload* is about the data uploading target (to a central server or an intermediate device).

Who: In this case, who refers to the participant executing the individual task (i.e., *who senses, who computes, and who uploads*). For simple MCS tasks, each participant just completes the sensing, computing, and data uploading independently. However, for some complicated MCS tasks, participant collaboration in sensing, computing, and uploading could be introduced to reduce the required resources, energy consumption, and cost of each individual device. Thus, a participant's individual task (or a sub-stage of sensing, computing, and uploading) might be offloaded to another participant or completed with another participant's assistance. On the other hand, for the sake of user privacy, how to anonymize *who* is also important.

How: Here, this is about developing mechanisms by addressing the above-mentioned 4W issues to fulfill the goals of saving energy, reducing mobile data costs, ensuring sensing coverage, and so on.

Table 2 gives a summary of the representative research work in *individual task execution*. Energy conservation is obviously one of the major issues in current MCS research. For instance, piggybacking crowd sensing [9] saves energy cost by exploiting other running applications for sensing/computing/uploading, while SociableSense [5] reduces energy consumption by adopting adaptive sensing cycles (*when to sense*) and computation offloading (*where to compute*) mechanisms. In addition to saving energy, some work also considers the sensing data quality by introducing the coverage constraint. For instance, [13] designs an energy-efficient collaborative mechanism for location-centric crowd sensing to ensure that each sensing area is covered by at

least one participant (*where to sense*). Besides, some work can help participants save mobile data cost as well as energy cost, using mechanisms such as data compression [14] and opportunistic free data uploading (via Bluetooth, WiFi) [15]. While data compression mainly considers *what to compute* and *what to upload*, opportunistic free data uploading often chooses the right timing and route for data uploading (*when to upload*). Furthermore, data uploading via relays by nearby devices (*where to upload*) or other participants (*who to upload*) might also reduce both energy consumption and mobile data cost [15].

4W1H IN CROWD DATA INTEGRATION

After the participants upload their personal sensed data, the central server will be responsible for *crowd data integration*. In this stage, the main issue is *to achieve the MCS task goals with collected sensed data*. In general, crowd data integration includes the whole process of data storage, data mining, data dissemination and visualization. The 4W1H issues in this stage include:

What: It refers to what tools and mechanisms can facilitate data storage, mining, and dissemination.

- Different data storage mechanisms are required to manage different kinds of sensed data. For simple data such as locations, numeric values, and enumeration items, traditional databases with spatial extension (e.g. *PostGIS* [4, 8]) are enough. For images and audios, a file storage system is recommended (e.g. *Amazon S3* [10]).

- The data mining process is crucial for crowd data integration as it bridges the gap between low-level personal sensed data and high-level intelligence. In general, data mining methods are MCS task-specific. For example, PEIR [4] generates personal environment influence reports for each participant by aggregating each participant's sensed personal data (e.g., GPS and activity) and environmental data (e.g., weather and road traffic), while *CrowdSense@Place* [3] attempts to classify unknown places by mining the participants' opportunistically captured images and audio clips.

• Integration result dissemination is related to user experiences. In addition to answering simple *user queries*, widely used mechanisms include *digital map visualization* [4, 8, 10], *social network sharing* [4], and *user feedback/alert* [5].

When: Here, this refers to the timing of crowd data integration. We can either process the streamed data in real time or manage the collected data afterward. For example, PEIR [4] and SociableSense [5] need (nearly) real-time data processing and use high-performance *online* algorithms to aggregate crowd data, while CrowdSense@Place [3] leverages *offline* algorithms to analyze the crowd data.

Who: We consider who as people who can benefit from the crowd data integration results. Apparently, there are two types of beneficiaries in this stage:

• The MCS participants who can get personalized services [4, 5, 10]

• The general public who can access the MCS results for whatever purpose [3]. While providing participants with personalized services could encourage people to participate in MCS tasks, MCS task serving for the general public is often influential and valuable for society.

Where: Traditionally, the MCS organizer develops a central server from scratch. As cloud computing and platform as a service (PaaS) technology become popular, an alternative approach is leveraging existing web services such as AWS to build a cloud central server. For example, the on-the-move project [10] uses *Amazon SimpleDB*, *S3*, and *EC2* together to create the server-side crowd data integration environment.

How: The key issue in crowd data integration is to fulfill the MCS task objectives by processing the sensed data from participants. Thus, how refers to the combination of mechanisms that handle the above 4W issues; for instance, *how to protect each participant's privacy when processing the sensed data from individuals* and *how to improve the data integration efficiency if real-time data processing is required*.

FUTURE MCS RESEARCH DIRECTIONS

With the 4W1H issues elaborated on in previous sections, it can be seen that while task allocation, user selection/recruitment, and MCS task design in the two middle stages will continue to be the hot research topics taking into account factors such as energy, cost, coverage, privacy, and data quality, some research issues are still not well addressed in the task creation and crowd data integration stages of the MCS life cycle. Based on the identified gaps, we foresee the following research directions in MCS task creation and crowd data integration.

Allowing the general public to be MCS organizers: (*who in task creation*) Currently, almost all the MCS organizers are big institutions and entities due to the high running cost and huge efforts involved. With the development of software tools and platforms for MCS task creation, the general public is expected to create MCS tasks easily for both personal objectives

and public purposes. For example, if a mother loses her child in a large supermarket, she could create an MCS task in real time that collects images and other information to help find the child quickly.

Creating an MCS task for emergency situations: (*when in task creation*) Most of the current MCS applications fall into the long-term MCS category, such as collective behavior tracking and environment monitoring. Little work has been done to leverage MCS tasks dealing with emergency events. For example, a criminal escapes in a red car after robbing a bank. If the police create an MCS task in real time to invite people around that bank to capture photos of any red cars encountered, it would probably help the police to find the criminal rapidly.

Offering both personalized and general results: (*who in crowd data integration*) Current MCS applications focus mainly on serving only one type of user: either providing individual participants personalized results, or providing the public general-purpose results. As these two types of services have their own advantages (*personalized results attract users to participate* vs. *general-purpose results benefit more people*), the MCS organizer should consider offering both of them. For example, SociableSense provides the general public with non-personalized rules for improving social relationships based on personalized suggestions for participants.

Scaling "real-time" MCS applications to large scope: (*when in crowd data integration*) Although many MCS frameworks claim that they can offer near-real-time crowd data processing in the central server, they usually lack thorough evaluation and detailed descriptions of how close to "real time" the MCS application can get, especially when the deployment scale is large. To this end, there is obviously an urgent need to design scalable MCS tasks and evaluate the system through large-scale deployment.

CONCLUSION

With the recent explosion of sensor-rich smartphones in the market, mobile crowd sensing has become a popular research and application field, and it is expected that the scope and depth of MCS research will further expand in the years to come. In order to sort out the research issues in this emerging research domain, in this article, we propose a four-stage life cycle for the MCS process (i.e., *task creation*, *task assignment*, *individual task execution*, and *crowd data integration*) and attempt to use 4W1H (i.e., *what*, *when*, *where*, *who*, and *how*) to characterize the research issues for each stage of the MCS life cycle. By introducing the notion of a four-stage MCS life cycle and 4W1H framework, most of the research issues in MCS are revealed in a structured manner, and several research directions are pointed out, aiming to provide insights and guidelines for young researchers in the MCS field.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (Nos. 61332005, 61373119).

After the participants upload their personal sensed data, the central server will be responsible for crowd data integration. In this stage, the main issue is to achieve the MCS task goals with collected sensed data. In general, crowd data integration includes the whole process of data storage, data mining, data dissemination, and visualization.

By introducing the notion of four-stage MCS lifecycle and 4W1H framework, most of the research issues in MCS are revealed in a structured manner and several research directions are pointed out, aiming to provide insights and guidelines for young researchers in the MCS field.

REFERENCES

- [1] R. K. Ganti, F. Ye, and H. Lei, "Mobile Crowdsensing: Current State and Future Challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, Nov. 2011, pp. 32–39.
- [2] A. T. Campbell et al., "The Rise of People-Centric Sensing," *IEEE Internet Computing*, vol. 12, no. 4, July–Aug. 2008, pp. 12–21.
- [3] Y. Chon et al., "Understanding the Coverage and Scalability of Place-Centric Crowdsensing," *Proc. UbiComp*, 2013, pp. 3–12.
- [4] M. Mun et al., "Peir, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research," *Proc. MobiSys*, 2009, pp. 55–68.
- [5] K. K. Rachuri et al., "SociableSense: Exploring the Tradeoffs of Adaptive Sampling and Computation Offloading for Social Sensing," *Proc. MobiCom*, 2011, pp. 73–84.
- [6] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment Framework for Participatory Sensing Data Collections," *Proc. Pervasive*, 2010, pp. 138–55.
- [7] M. R. Ra et al., "Medusa: A Programming Framework for Crowd-Sensing Applications," *Proc. MobiSys*, 2012, pp. 337–50.
- [8] G. Cardone et al., "Fostering Participation in Smart Cities: A Geo-Social Crowdsensing Platform," *IEEE Commun. Mag.*, vol. 51, no. 6, June 2013, pp. 112–19.
- [9] N. D. Lane et al., "Piggyback Crowdsensing (PCS): Energy Efficient Crowdsourcing of Mobile Sensor Data by Exploiting Smartphone App Opportunities," *Proc. SenSys*, 2013, article no. 7, 14 pages.
- [10] W. Sherchan et al., "Using On-the-Move Mining for Mobile Crowdsensing," *Proc. MDM*, 2012, pp. 115–24.
- [11] H. Xiong, L. Wang, and D. Zhang, "Eemc: An Energy-Efficient Mobile Crowdsensing Mechanism by Reusing Call/SMS Connections," *Proc. NetMob*, 2013, pp. 323–29.
- [12] D. Yang et al., "Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing," *Proc. MobiCom*, 2012, pp. 173–84.
- [13] X. Sheng, J. Tang, and W. Zhang, "Energy-Efficient Collaborative Sensing With Mobile Phones," *Proc. INFOCOM*, 2012, pp. 1916–24.
- [14] K. C. Barr and K. Asanovic, "Energy-Aware Lossless Data Compression," *ACM Trans. Computer Systems*, vol. 24, no. 3, Aug. 2006, pp. 250–91.
- [15] L. Wang, D. Zhang, and H. Xiong, "Effsense: Energy-Efficient and Cost-Effective Data Uploading in Mobile Crowdsensing," *Proc. UbiComp Adjunct*, 2013, pp. 1075–86.

BIOGRAPHIES

DAQING ZHANG (daqing.zhang@telecom-sudparis.eu) is a professor at TELECOM SudParis, France. His research interests include big data analytics, urban computing, context-aware computing, and ambient assistive living. He is an Associate Editor for *ACM Transactions on Intelligent Systems and Technology*. He has been a frequent Invited Speaker at various international events. He is the winner of the Ten Years CoMoRea Impact Paper Award at IEEE PerCom 2013, and Best Paper Awards at IEEE UIC 2012 and Ubiquitous 2011.

LEYE WANG (leye.wang@telecom-sudparis.eu) is currently a Ph.D. student at Institut Mines-TELECOM/TELECOM SudParis and Université Pierre et Marie Curie. He received his M.Sc. and B.Sc. in computer science from Peking University. His research interests include mobile crowdsensing and ubiquitous computing.

HAOYI XIONG (haoyi.xiong@telecom-sudparis.eu) is currently a Ph.D. student at Institut Mines-TELECOM/TELECOM SudParis and Université Pierre et Marie Curie. He received his M.Sc. in information technology from the Hong Kong University of Science and Technology. His research interests include mobile crowdsensing and ubiquitous computing. He received the Best Paper Award from IEEE UIC 2012.

BIN GUO (guob@nwpw.edu.cn) is a professor in the School of Computer Science, Northwestern Polytechnical University, China. He received his Ph.D. degree from Keio University, Tokyo, Japan, in 2009. During 2009–2011, he was a post-doctoral researcher at Institut TELECOM SudParis in France. His research interests include pervasive computing, social computing, and mobile crowd sensing. He has served as an Associate/Guest Editor for a number of international journals, such as *IEEE Communications Magazine*, *Personal and Ubiquitous Computing*, *ACM Transactions on Intelligent Systems and Technology*, and others. He has served as the General/Program/Workshop Chair for several conferences, including IEEE SCI@PerCom, IEEE UIC, and IEEE IThings.