symmetry

MDPI

Article

Task-Management Method Using R-Tree Spatial Cloaking for Large-Scale Crowdsourcing

Yan Li and Byeong-Seok Shin *

Department of Computer Engineering, Inha University, #100 Inha-ro, Nam-gu, Incheon 22212, Korea; leeyeon@inha.ac.kr

* Correspondence: bsshin@inha.ac.kr; Tel.: +82-032-860-7452

Received: 30 September 2017; Accepted: 7 December 2017; Published: 10 December 2017

Abstract: With the development of sensor technology and the popularization of the data-driven service paradigm, spatial crowdsourcing systems have become an important way of collecting map-based location data. However, large-scale task management and location privacy are important factors for participants in spatial crowdsourcing. In this paper, we propose the use of an R-tree spatial cloaking-based task-assignment method for large-scale spatial crowdsourcing. We use an estimated R-tree based on the requested crowdsourcing tasks to reduce the crowdsourcing server-side inserting cost and enable the scalability. By using Minimum Bounding Rectangle (MBR)-based spatial anonymous data without exact position data, this method preserves the location privacy of participants in a simple way. In our experiment, we showed that our proposed method is faster than the current method, and is very efficient when the scale is increased.

Keywords: spatial cloaking; R-tree; large scale; spatial crowdsourcing

1. Introduction

With the development of sensor and data communication technologies, smart devices can collect real-time environmental information. Thus, the paradigm of information systems is changing from the existing service center to the data center. As real-time data and data analysis have become important in the data-driven paradigm, various methods of data collection have been studied in recent research [1].

In existing sensor networks, when collecting real-time environmental data, we must deploy many sensors in the area where the measurement is needed. The disadvantage of the traditional sensor network method is that the maintenance cost for the sensor network is high. A crowdsourcing system can collect the necessary data directly from volunteer participants with smart devices and deliver it to a crowdsourcing server. This makes it easy to collect data once or over a short period with little expense and greater accuracy compared to current sensor networks [2].

Spatial crowdsourcing collects sensed information using smart devices equipped with GPS(global positioning system) and other location-detection sensors. As a result, most crowdsourcing results contain location data, and the collected results can be analyzed from a map-based perspective focusing on local spatial characteristics. The visibility of analysis results can also be improved in a spatial crowdsourcing system.

Adequate user participation is one of the most critical factors in the service quality of spatial crowdsourcing applications. Most of the current crowdsourcing applications [3–8] are based on voluntary participation. When participants are selected as attendees, they consume their own resources, such as battery and computing power, and expose their locations, leading to potential privacy threats. Zhao, Li, and Ma [9] studied general participant selection problem-solving solutions. They defined the participants' consumption model and the value model of participants' contributions in order to support the selection of a set of participants for task registers based on supplier or company. In this system, the task assignment is based on the balance of a participant's consumption and the service providers' discharge.

In our work, we consider task assignment to be the main challenge in a crowdsourcing system; other challenges include the incentive mechanism, the absence of real-world datasets, and privacy protection. The spatial crowdsourcing server assigns tasks to numerous participants based on real-time costs [10,11]. We only consider systems that assigns requesters' tasks to participants automatically. Automatic task assignments minimize participants' operation costs while preserving their location privacy, because in this case the participants don't need to care about what kind of tasks they can approve or where they have to go.

In this paper, we propose an R-Tree spatial cloaking-based task assignment method as a spatial crowdsourcing system. We use MBR data instead of location-point data to assign tasks to participants. When the participants are close to the MBR boundary, data sensing automatically starts; it lasts until the participants are within the MBR boundary. The whole process could be carried out automatically, so it would be able to attract more participants and could increase the amount of crowdsensing data. When the results are translated to the crowdsourcing server, it sends data package lists, including location data, time data, and sensing data. To preserve participants' location privacy, the local position relative to the MBR is used in this data package. The MBR data is sent in the first data package only. This proposed method would increase the amount of sensing data while also preserving participants' location privacy. The MBR data would be used R-tree-based data storage in the crowdsourcing server to reduce the data insertion cost.

This paper is organized into five sections. In the second section, we describe the current related research on spatial crowdsourcing and task assignment in spatial crowdsourcing systems. In the third section, we introduce our proposed method: the R-tree spatial cloaking-based task assignment method. The fourth section describes our experimental results. The last section presents our conclusions and lays out several plans for future work.

2. Related Works

Spatial crowdsourcing is a type of online, location-based crowdsourcing that asks participants to physically travel to the specific location of a task during the term of the task. There are two types of spatial crowdsourcing: in the first, participants choose tasks from a server; in the second, the server automatically assigns spatial tasks to participants who have registered their location on the server. In this paper, only automatic task-management algorithms of the second type of spatial crowdsourcing are used [2].

Currently, there are several general spatial crowdsourcing platforms, including gMission and GeoCrowd, that support various user-defined spatial crowdsourcing application services. These general platforms model spatial tasks for a specific service, allowing spatial tasks to be propagated to normal smart device users. Sensing results are then collected in the crowdsourcing server [12].

Mobile crowdsourcing systems can collect real-time environmental data from smart devices, and the collected data can be used for end-user services online. The end users use sensed environmental data along with other attributes such as position or time data. When the participants' personal data, including position data, are open to a third party, their position privacy has been attacked. Our research focuses on the protection of position data as they are transmitted between participants and the mobile crowdsourcing server [13].

The location privacy problem emerges when data are sent between the crowdsourcing server and participants to assign tasks and provide results. Some recent studies have proposed a task assignment method that is limited to participants' area. In addition, some researchers have studied methods of protecting participants' location privacy using encryption algorithms.

Several studies have tried to solve the privacy problem using the trusted encryption system framework. These have looked at numerous research challenges, including the computational efficiency of allocation and payment, the individual rationality of each participant, budget feasibility, truthfulness, consumer sovereignty, and the competitiveness of participants. Once the encryption system framework has been compromised, however, location information cannot be protected, and the framework is easy for fraudulent users to access. Moreover, by combining several participants' information, fraudulent users can easily guess a specific participant's approximate location. Recent studies that are closely related to ours include those by Ul Hassan and Curry [14] and Tong, She, Ding, Wang, and Chen [15]. Both studies consider the online spatial task assignment problem; however, they differ from our work in terms of their problems and objectives.

A location k-anonymity-based method has been proposed to support personal privacy by preserving mobile-based LBS service by Bugra and Ling [16]. Under this method, a personalized k-anonymity model was proposed that can modify the level of anonymity of the mobile user. The location accuracy could be decreased through enlarging the spatial area exposed for a given user. This kind of method is called spatial cloaking. The spatial cloaking method is used to blur participants' exact location data and preserve location privacy [17]. Pournajaf, Xiong, Sunderam, and Goryczka [18] proposed a tow-stage optimization approach to manage location uncertainty when using cloaked location data. This method uses a round area for spatial cloaking and point data, and thus could increase the computing cost from the server side. Ul Hassan and Curry [14] studied the combinatorial fractional optimization approach for efficient task assignment. They used the semi-bandit learning method to reduce participants' travel costs [19]. They called this method the Distance-Reliability Ratio (DRR) algorithm. The DRR algorithm computes all distances between participants and target locations online, which can increase computing costs. This method does not consider the participants' location privacy during crowdsourcing [13,20].

More recently, differential geo-obfuscation mechanisms have been studied for guaranteeing the location privacy of location-based services [21]. The study by Leye Wang et al. [22] introduced a location privacy-preserving task-allocation method by using differential geo-obfuscation. In this method, they use participant's obfuscated locations when allocating crowdsensing tasks to participants. Geo-obfuscation-aware methods have to recalculate each real location point into obfuscation-aware points when allocating tasks to participants. However, the collected location-dependent data have to be calculated into real data first in order to guarantee the accuracy of the crowdsensing results. Therefore, it would increase the cost of data management of scalable crowdsourcing server-side systems.

The scalability of crowdsourcing systems has also been studied for several years [23]. A study by Oleson et al. [24] introduced a scalable quality-control method for crowdsourcing. In this study, the authors introduced cost-based scalability for crowdsourcing systems in order to ensure that more workers would be able to be hired. Van Pelt and Sorokin studied a method for filtering out bad contributions to reduce server-side costs and increase the quality, cost, and scalability of the crowdsourcing system [25]. We use the R-tree index [26] to manage spatial crowdsourcing tasks and to reduce the server cost and increase the scalability of the crowdsourcing system.

3. R-Tree Spatial Cloaking-Based Task Assignment Method

We studied an R-Tree spatial cloaking-based task-assignment method for spatial crowdsourcing systems, especially for automatically assigning tasks to participants.

3.1. Spatial Task Management Process

Crowdsourcing data are used for several types of service in open crowdsourcing platforms [27,28]. For example, real-time spatial crowdsourcing data can be used for online monitoring systems for end users, and historical spatial crowdsourcing data can be used for various types of map-based analytics services [29,30]. To support these types of map-based services, we must transform spatial task results into location-based report data as quickly as possible. However, in this process, we must also protect participants' location privacy. Thus, we propose a grid-based index method to improve the spatial data insert speed and the use of grid-based encryption of location data during the spatial task-management process to safeguard participants' location privacy. The proposed spatial task-management process is shown in Figure 1.

On a crowdsourcing platform, task requesters must register the type of data resources they need in a specific area on a centralized spatial crowdsourcing server. The server then assigns tasks to participants. Each task includes available date and time with area information and target data type. The reward for task processing is also included; but in our research, we will not consider the benefit for participants. If the task area is very large, then the area will be divided into subtasks. When a participant accepts the assigned task, he or she travels to the task location; upon reaching the area, the location-based task is completed. The collected resources are then sent to the centralized crowdsourcing server.

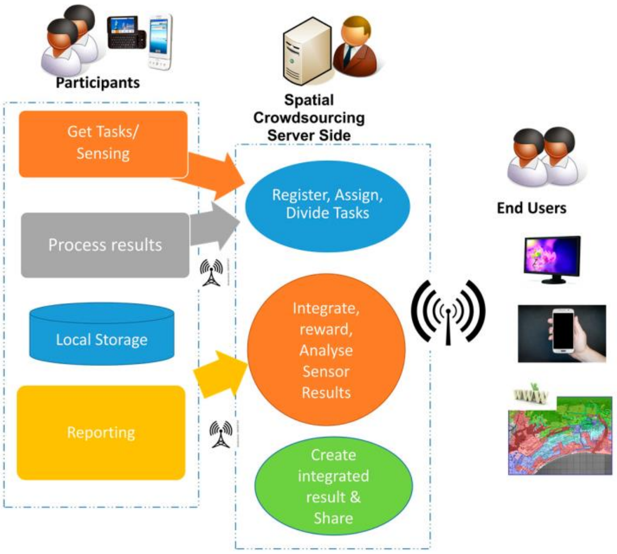
Participants
Get Tasks/ Sensing
Process results
Local Storage
Reporting
Spatial Crowdsourcing Server Side
Register, Assign, Divide Tasks
Integrate, reward, Analyse Sensor Results
Create integrated result & Share
End Users

Figure 1. Proposed spatial task-management process.

When participants accept tasks from the centralized crowdsourcing server, the regional information and participants' data is added to the crowdsourcing server during a participant registration process. Following this, when a participant is in the sensing region, the R-tree based index inserts his or her data into the crowdsourcing server.

A spatial task includes spatial area S (a rectangle-based area including the Minimum Bounded Rectangle(MBR) of R-tree), task execution time information T ($t1 < T < t2$), and spatial query Q; this type of spatial task is converted into a task map, shown as TM = {S, T, Q}. When the task requester registers the TM on the crowdsourcing server, the crowdsourcing platform assigns the task to participants in the form of a geometric-based task. The participants send results in the form of RM = {participant's ID, time, task ID, location-based result}. Our goal is to design an online task-management method to ensure the computational efficiency of the crowdsourcing server and the participants' location privacy. To protect participants' location privacy and ensure the server's scalability, the location data used in the RM is encrypted using the grid-based index proposed in Section 3.2.

3.2. R-Tree-Based Task-Assignment Method

We propose an R-tree spatial cloaking-based task-assignment method for spatial crowdsourcing systems. We assume that the requesters' tasks are registered on the crowdsourcing server and assigned to preregistered participants automatically. The goal of our research is to reduce the computing costs of the crowdsourcing server while preserving the location privacy of participants.

The information that requesters need from the crowdsourcing server includes the crowdsourcing area, duration of collection time, and density of sensing data. Then, the crowdsourcing server constructs an estimated R-tree based on the density data and crowdsourcing area. The central server assigns tasks to registered participants by using the MBR of the R-tree leaf node. Our method uses MBR data instead of location-point data.

When participants get close to the MBR boundary, the data sensing ceases and the target data is automatically sensed. This continues until the participants leave the MBR boundary. The sensed data is organized as a data package list and then sent to the crowdsourcing server. The first data package includes the MBR data, while the other data package uses only the anonymous local position relative to the MBR to preserve participants' location privacy. When the MBR is expressed as [(x_1, y_1), (x_2, y_2)], and use (*Ap(Anonymous position))* to simply the anonymous position data, the anonymous location position of MBR is calculated by using the following formula. In this formula ∂ is defined by the density of participants. When the density of destination MBR is low the ∂ will be large, if the density of destination MBR is high then the ∂ will be small. In any case, this kind of location anonymity could affect the correctness of the crowdsourced result. However, when the information sought by crowdsourcing is an area-based request, such as the collection of noise in specific area, and then the effectiveness of this kind of location anonymity is very poor.

$$Ap(x',y') = [\left\lfloor \frac{\partial(x-x_1)}{x_2-x_1} \right\rfloor * \frac{(x_2-x_1)}{\partial}, \left\lfloor \frac{\partial(y-y_1)}{y_2-y_1} \right\rfloor * \frac{(y_2-y_1)}{\partial}]$$

When the first data package is inserted into the server, the server finds the leaf node of the R-tree to insert the data. If the server receives a participant's result data that is the same as position data, then the data package is considered to be one of the sensing results of the previous MBR node. This data package could be directly inserted into the leaf node without any more R-tree pruning to reduce the insertion cost.

In some cases, the task area S to be managed will be very large, and then the task will be separated into several sub-tasks, withthe area S being broken down into several sub areas S_1,S_2,S_3, etc. In this case, the main task will be the parent of a sub-tree, and the sub-tasks will be the child nodes of that sub-tree. This reduces the integration costs for the sensing results. The size of the task area is defined by the task registration, and the area is separated when the number of participants is greater than TA-Limitation. TA-Limitation is determined by the performance server computer. In our test we designated the bounds of TA-Limitation to be 10 < TA-Limitation < 100.

Figure 2 shows the architecture of the proposed R-tree spatial cloaking-based method. When the crowdsourcing request registers in the crowdsourcing server, the R-tree is estimated by using the sourcing area and density data. The task then selects preregistered participants; for example, the R7 task selected Participant 1, who is near the target area. The task ID and the selected participants will be managed separately. When the number of participants increases, the R-tree node will be separated. In our method, parent nodes will include taskIDs. At the pre-leaf node level, each parent node includes only one taskID to reduce the pruning time.

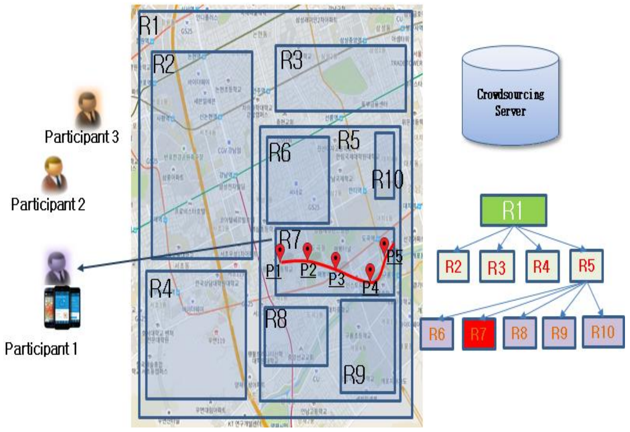

Figure 2. Magnetization as a function of applied field.

As the participants get close to the R7 area, the task automatically senses target data in the positions P1, P2, P3, P4, and P5. Using our method, the sensing results data are inserted into the R7 node directly, without any tree pruning. When the searching area S is so large, then S will be separated into several sub-areas automatically, using the traditional R-Tree method. This area will include the sub-areas for other tasks, too. In our method, however, the sub-tree of one task is managed independently. This method increases the amount of sensing data, preserves the location privacy of participants, and reduces the data insertion costs for the crowdsourcing server.

4. Experimental Results

We used a Gowalla [31] dataset and constructed a crowdsensing server with our proposed R-tree spatial cloaking-based method. A Gowalla dataset is a location-based dataset in which users share their locations with one another. This dataset contained 6,442,890 check-in location-based transaction data from the period of February 2009–October 2010. This dataset showed real human motion and dynamics. In our experiment, each Gowalla user was considered a participant. We simulated several rectangle area-based tasks, and emulated an entire task-management process, including task propagation, task assignment, and participants' sensor results. Then, we analyzed the data insert performance and memory usage of the server side.

We compared our R-tree-based method to the tow-stage optimization method, which uses an algorithm to assign tasks, and a base system that uses an R-tree method only in the server. For the random mechanism, we obtained the average performance of 50 such solutions for evaluations; in each solution, the density threshold was chosen at random from a range of 1 to 29 POIs(Point of Interest). All the simulations were run on a computerwith 1.7 GHz CPU and 8 GB memory. Each measurement was averaged over 100 instances.

In order to show the effect scalability of the proposed algorithm clearly, it is necessary to reduce the noise of other factor such as speed, so we assumed that the moving speed of participants was the same when they received tasks, and thus that this did not affect the result. We randomly simulated 100 to 900 participants from the Gowalla dataset to construct a crowdsensing environment and created 100 different rectangle-based tasks to randomly assign to participants at one time.

Figure 3 shows the results of our experiment. The R-tree spatial cloaking-based method performed faster than the tow-stage optimization method when the number of participants increased from 100 to 900 for 100 tasks. Because our method inserts the results into the R-tree without tree pruning, the management cost is reduced when the tow-stage optimization method inserts the results data without any data structure. When the density of MBR increased, the MBR will be separated under the stratagem of the R-tree to reduce the insertion bottle neck. Our experiment also showed that, when the number of participants was only 100, the running time for each method was similar.

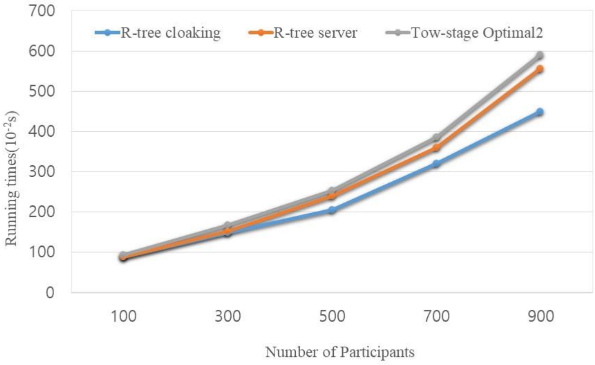

Figure 3. Rectangle-based crowdsourcing tasks and running time comparison.

Secondly, we tested the preconstruction cost and memory usage of our proposed method. The memory usage comparison result is shown in Figure 4. In the previous running time test, we also measured the memory usage of the three methods. As Figure 4 shows, when the number of participants was same, and using the same tasks to process the crowd sourcing, the proposed R-tree cloaking method used little more memory than the traditional R-tree and tow-stage optimization methods. Because our method has to store the relation of task and subtask separately, the R-tree cloaking method has a greater cost. Although the proposed method uses almost 3% more memory than the tow-stage method, for large-scale crowdsourcing systems, real-time processing is more important than memory usage. Therefore, the proposed method is efficient for large-scale crowdsourcing systems.

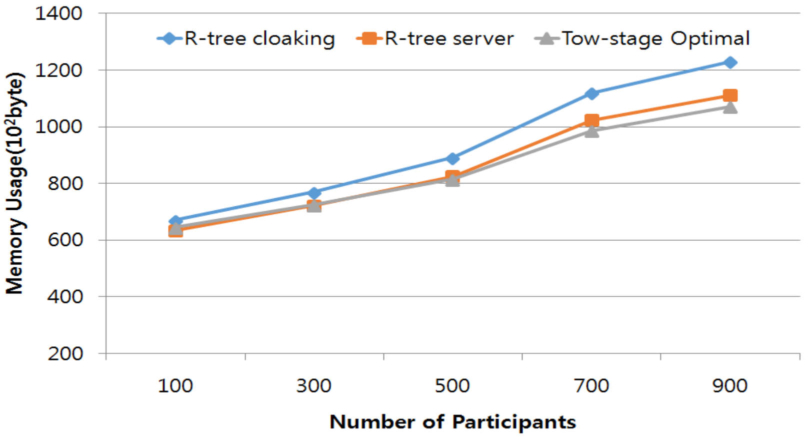

Figure 4. Rectangle-based crowdsourcing tasks and memory usage comparison.

Thirdly, we tested the prevention of center-of-area attack. We compared our proposed method with the tow-stage optimization method. We used the 900 generated participants and simulated for 5 min, then we divided the test area into 16 regions. One of the separated regions was selected for center-of-area attack. We tried to evaluate the effectiveness of privacy prevention against center-of-area attack by measuring the distance between the query issuer and the center of the cloaking region. Figure 5 shows the results of three different regions. Because each test region had a different size, the distance value to the center of the cloaking region was normalized to 0–100. The proposed method and tow-stage optimization methods were both able to separate the center of the region away from the real query area; but the distance of the proposed method is more random than that of the tow-stage optimization method.

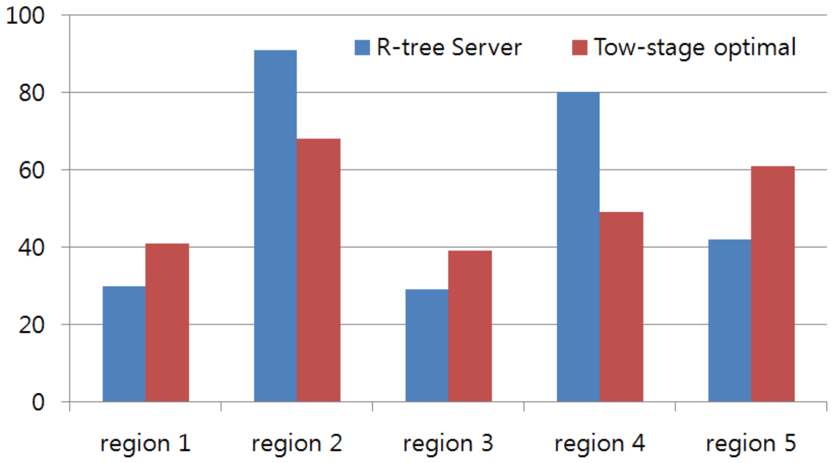

Figure 5. Center-of-area attack prevention comparison.

5. Conclusions

We studied an R-tree spatial cloaking-based task-assignment method for use in spatial crowdsourcing. With the popularization of the data-driven service paradigm, the question of how to efficiently collect real-time data has become a hot research topic. Crowdsourcing using participants' smart devices is a new way to collect real-time data online. Privacy is a serious concern when engaging participants in spatial crowdsourcing. We used the anonymous MBR data of an estimated server-side R-tree instead of exact location-point data in order to maintain participants' privacy. The anonymous MBR data was used for R-tree-based data storage in the crowdsourcing server to reduce the data insertion cost with the least reduction in the accuracy of the crowdsourced results. Our proposed method was able to reduce the insertion cost when the crowdsourcing system is scaled up. In the future, we need to test the preconstruction cost of different methods and study the multi-task execution method.

Acknowledgments: This research was supported by INHA University Research Grant.

Author Contributions: Yan Li and Byeong-Seok Shin designed all of the experiments together. Yan Li performed the range-based R-tree cloaking algorithms. Byeong-Seok Shin designed experimental process and platform. Then, Yan Li developed and produced the test program to conduct the experiments on the proposed method. The traditional R-tree and two-stage optimization methods were developed by Byeong-Seok Shin.

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

References

1. Tran, L.; To, H.; Fan, L.; Shahabi, C. A real-time framework for task assignment in hyperlocal spatial crowdsourcing. *ACM Trans. Intell. Syst. Technol.* **2017**.
2. To, H. Task Assignment in Spatial Crowdsourcing: Challenges and Approaches. In Proceedings of the 3rd ACM SIGSPATIAL PhD Symposium, San Francisco, CA, USA, 26 June–1 July 2016.
3. Sensorly. Available online: http://www.sensorly.com (accessed on 20 May 2017).
4. Koukoumidis, E.; Peh, L.-S.; Martonosi, M.R. Signalguru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory. In Proceedings of the ACM MobiSys, Bethesda, MD, USA, 28 June–1 July 2011; pp. 127–140.
5. Mohan, P.; Padmanabhan, V.N.; Ramjee, R. Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones. In Proceedings of the ACM SensSys, Raleigh, NC, USA, 5–7 November 2008; pp. 323–336.
6. Thiagarajan, A.; Ravindranath, L.; LaCurts, K.; Madden, S.; Balakrishnan, H.; Toledo, S.; Eriksson, J. Vtrack: Accurate, Energy-aware Road Traffic Delay Estimation Using Mobile Phones. In Proceedings of the ACM SenSys, Berkeley, CA, USA, 4–6 November 2009; pp. 85–98.
7. Rana, R.K.; Chou, C.T.; Kanhere, S.S.; Bulusu, N.; Hu, W. Earphone: An End-to-end Participatory Urban Noise Mapping System. In Proceedings of the ACM/IEEE IPSN, Stockholm, Sweden, 12–16 April 2010; pp. 105–116.
8. Stevens, M.; Hondt, E.D. Crowdsourcing of Pollution Data Using Smartphones. In Proceedings of the ACM UbiComp, Copenhagen, Denmark, 26 September 2010; pp. 1–4.
9. Zhao, D.; Li, X.-Y.; Ma, H. How to Crowdsource Tasks Truthfully without Sacrificing Utility: Online Incentive Mechanisms with Budget Constraint. In Proceedings of the IEEE, Toronto, ON, Canada, 27 April–2 May 2014.
10. Singer, Y.; Mittal, M. Pricing Mechanisms for Crowdsourcing Markets. In Proceedings of the World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 1157–1166.
11. Singla, A.; Krause, A. Truthful Incentives in Crowdsourcing Tasks Using Regret Minimization Mechanisms. In Proceedings of the World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 1167–1178.
12. Kazemi, L.; Shahabi, C. Geocrowd: Enabling Query Answering with Spatial Crowdsourcing. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 6–9 November 2012.
13. To, H.; Ghinita, G.; Shahabi, C. A Framework for Protecting Worker Location Privacy in Spatial Crowdsourcing. *Proc. VLDB Endow.* **2014**, *7*, 919–930. [CrossRef]

14. Hassan, U.; Curry, E. Efficient task assignment for spatial crowdsourcing: A combinatorial fractional optimization approach with semi-bandit learning. *Expert Syst. Appl.* **2016**, *58*, 36–56. [CrossRef]
15. Tong, Y.; She, J.; Ding, B.; Wang, L.; Chen, L. Online Mobile Micro-Task Allocation in Spatial Crowdsourcing. In Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering, Helsinki, Finland, 16–20 May 2016; pp. 49–60.
16. Bugra, G.; Liu, L. Location privacy in mobile systems: A personalized anonymization model. In Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, Columbus, OH, USA, 6–10 June 2005.
17. Chow, C.-Y.; Mokbel, M.F.; Liu, X. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. *Geoinformatica* **2011**, *15*, 351–380. [CrossRef]
18. 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, Brisbane, Australia, 14–18 July 2014; pp. 73–82.
19. 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. [CrossRef]
20. Cheng, P.; Lian, X.; Chen, Z.; Fu, R.; Chen, L.; Han, J.; Zhao, J. Reliable Diversity-based Spatial Crowdsourcing by Moving Workers. *Proc. VLDB Endow.* **2015**, *8*, 1022–1033. [CrossRef]
21. Gardner, Z.; Leibovici, D.; Basiri, A.; Foody, G. Trading-off location accuracy and service quality: Privacy concerns and user profiles. In Proceedings of the International Conference on Location and GNSS, ICL-GNSS 2017 Conference, Nottingham, UK, 27–29 June 2017.
22. Wang, L.; Yang, D.; Han, X.; Wang, T.; Zhang, D.; Ma, X. Location Privacy-Preserving Task Allocation for Mobile Crowdsensing with Differential Geo-Obfuscation. Proceeding of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 627–636.
23. Alfarrarjeh, A.; Emrich, T.; Shahabi, C. Scalable Spatial Crowdsourcing: A Study of Distributed Algorithms. In Proceedings of the 16th IEEE International Conference on Mobile Data Management, Pittsburgh, PA, USA, 15–18 June 2015.
24. Oleson, D.; Sorokin, A.; Laughlin, G.; Hester, V.; Le, J.; Biewald, L. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Hum. Comput.* **2011**, *11*, 43–48.
25. Van Pelt, C.; Sorokin, A. Designing a Scalable Crowdsourcing Platform. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, Scottsdale, AZ, USA, 20–24 May 2012.
26. Guttman, A. R-trees: A dynamic index structure for spatial searching. In Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, Boston, MA, USA, 18–21 June 1984.
27. Cheng, P.; Lian, X.; Chen, L.; Han, J.; Zhao, J. Task assignment on multi-skill oriented spatial crowdsourcing. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 2201–2215. [CrossRef]
28. To, H.; Ghinita, G.; Fan, L.; Shahabi, C. Differentially private location protection for worker datasets in spatial crowdsourcing. *IEEE Trans. Mob. Comput.* **2017**, *16*, 934–949. [CrossRef]
29. Deng, D.; Shahabi, C.; Demiryurek, U.; Zhu, L. Task selection in spatial crowdsourcing from worker's perspective. *Geoinformatica* **2016**, *20*, 529–568. [CrossRef]
30. Pournajaf, L.; Xiong, L.; Sunderam, V. Dynamic data driven crowd sensing task assignment. *Procedia Comput. Sci.* **2014**, *29*, 1314–1323. [CrossRef]
31. Cho, E.; Myers, S.A.; Leskovec, J. Friendship and mobility: User movement in location-based social networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011.