User Recruitment System for Efficient Photo Collecting in Mobile Crowdsensing

En Wang, Yongjian Yang[†], Jie Wu, *IEEE Fellow*, Kaihao Lou, Dongming Luan, and Hengzhi Wang

***Abstract*—Mobile crowdsensing recruits a group of mobile users to cooperatively perform a common sensing job with their smart devices. As a special issue, photo crowdsensing allows users to utilize the built-in cameras of mobile devices to take photos for an event or a target. Then, the photos can be used in numerous application areas, such as target reconstruction, scenario reduction and so on. Therefore, photo crowdsensing has attracted considerable attention recently due to the rich information that can be provided by images. In this paper, we focus on using the photos to make reconstructions for the specific targets. Furthermore, we develop a user recruitment system for efficient photo collecting in mobile crowdsensing (RSMC), where the task requesters publish a sensing task to the users, and the map is gridded according to the locations of the sensing targets. Then, we use a semi-Markov model to calculate the user's utility for the sensing task. Finally, a user recruitment strategy is devised to recruit the optimal k users for finishing the sensing task. We conduct extensive simulations based on three widely used real-world traces: *roma/taxi*, *epfl* and *geolife*. The results show that compared with other recruitment strategies, RSMC takes the largest number of efficient photos for the sensing task.**

***Index Terms*—Mobile crowdsensing, User recruitment, Photo collecting, Semi-Markov**

I. Introduction

The rising popularity of smartphones has led to a new world consisting of phones, and they have been equipped with a variety of sensors (e.g., camera, light sensor, chemical sensor and GPS) that allow them to be considered mobile devices with a powerful sensing ability. Due to these developments, a new sensing paradigm called mobile crowdsensing (MCS) has been proposed [1] to recruit some mobile users for finishing a common sensing task through their smartphones. The collection and processing of sensing data on mobile users' smartphones produce useful knowledge, which serves many applications ranging from traffic route planning and available resources reporting to specific target tracking and air quality monitoring [2–8].

In this paper, we are particularly interested in mobile visual crowdsensing (MVC) [9], [10], where people take photos for an interesting event or a specific target with their mobile devices. Compared with the traditional MCS, MVC collects richer information through images. Hence, in many cases, MVC is superior to traditional crowdsensing that collects sensing data through deploying fixed sensors.

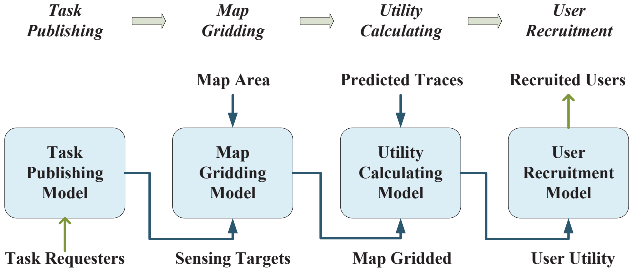


Fig. 1. User recruitment system framework in mobile crowdsensing. The framework is composed of the following four models: task publishing, map gridding, utility calculating and user recruitment models.

Previous works in MVC mainly include two aspects. The first aspect is developing some applications, where Creek-Watch [11], PhotoNet [12], PhotoCity [13], and WreckWatch [14] are some examples in which the built-in cameras of smart devices have been used to take photos for specific tasks. The second aspect focuses on data utility measurement or data collection of the photos taken by the workers [15]. The fact that few works propose user recruitment strategies in MVC, especially for developing user recruitment systems in MVC, is common knowledge. Actually, in MCS, the user recruitment problem has been discussed many times in mobile crowdsensing, and there has been much research on user recruitment strategies [16], [17], where a common challenge for most mobile crowdsensing applications is to identify mobile users who can contribute the most value to the sensing task. However, compared with the user recruitment in MCS, MVC faces several distinct challenges, such as multidimensional photo coverage needs and photo redundancy identification and elimination. Hence, proposing a user recruitment system for MVC is also necessary and urgent. To this end, in this paper, we develop a user recruitment system, where we focus on using the photos to make reconstructions for the targets of a specific area (e.g., make a reconstruction for a tourist spot or a city area). This approach raises the following research challenges.

- *Efficient Areas*: according to the specific target reconstruction task, we should decide the efficient areas for users to take photos, and it is useful to measure the efficiency of users, especially for taking the mobility traces into consideration.
- *User Utility*: according to the users' mobility traces and the efficient areas, we must decide the user's utility for the specific target reconstruction task.
- *Optimal User Set*: according to the users' utilities and

E. Wang, Y. Yang, K. Lou, D. Luan and H. Wang are with the Department of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China. (E-mails: wangen@jlu.edu.cn; yyj@jlu.edu.cn; loukh5513@mails.jlu.edu.cn; luandm5513@mails.jlu.edu.cn; wanghz5513@mails.jlu.edu.cn)

J. Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA. E-mail: jiewu@temple.edu.

Y. Yang[†] is the corresponding author

This work is supported by the National Natural Science Foundation of China under Grant No. 61772230 and No. 61702215, China Postdoctoral Science Foundation No. 2017M611322. This work is also supported in part by NSF grants CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, and CNS 1439672.

the recruiting budget, we should decide the top-k users to finish the crowdsensing task.

To solve the above challenging problems, we design a user recruiting system for efficient photo collecting in mobile crowdsensing, which consists of four models (as shown in Fig. 1):

- *Task Publishing Model*: a requester publishes a sensing task for taking photos of the specific targets. To recruit suitable users for finishing the sensing task, we divide the task into the locations of the target buildings or tourist spots. We also decide the efficient areas for users to take photos, according to the task locations.
- *Map Gridding Model*: according to the requester's sensing task, the positions of the target buildings or spots are decided, and we divide the map into grids. To map the targets into the specific grids, the surrounding (north, south, east, and west) grids of the targets are defined as the efficient areas. The target reconstruction can be accomplished by the photos taken around the targets.
- *Utility Calculating Model*: we use a semi-Markov model to determine the probability distribution of the user arrival time at the efficient areas, and then, the user's utility is determined by the expected number of efficient photos that they can take before the sensing task's deadline.
- *User Recruitment Model*: taking the redundant photos (photos taken in the same efficient area) into consideration, the top-k users recruiting problem is NP-hard. We use a submodular function to solve the NP-hard problem; then, the top-k users can be recruited to take the expected largest number of efficient photos.

The main contributions of this paper are briefly summarized as follows:

- We design a useful system framework (including task publishing, map gridding, utility calculating and user recruitment models) for recruiting the most suitable users to finish the crowdsensing tasks.
- We propose a semi-Markov model to determine the probability distribution of the user arrival time at the efficient grids and then obtain the utility function.
- We formulate the top-k user recruitment problem as an NP-hard problem, and we adopt a practical greedy heuristic that uses the submodular function to solve the NP-hard recruiting problem.
- We conduct extensive simulations based on three widely used real-world traces: *roma/taxi*, *epfl*, and *geolife*. The results show that compared with other recruitment strategies, RSMC takes the largest number of efficient photos for the sensing task.

The remainder of this paper is organized as follows: we review the related work in Section II. The user recruitment system for collecting photos in MCS is described in detail in Section III. In Section IV, we evaluate the performance of the designed recruitment system through extensive simulations. We conclude the paper in Section V.

II. Related Work

A. Visual Crowdsensing System

Focusing on automated scene analysis, Li *et al.* [18] conduct a crowded scene survey, which includes crowdsensing models, algorithms, protocols, and system performance. Chen *et al.* [19] propose a coadjutant visual sensing system through sharing pictures and creating a virtual opportunistic community. Guo *et al.* [20] also conduct a study in terms of the picture collection problem in mobile crowd photographing, and they propose a pyramid-tree model to collect the best subset of pictures. Wang *et al.* [21] utilize the combination of mobile crowd sensing and crowd smart to sense temporary obstacles and send timely reminders to the road walkers. Dong *et al.* [22] propose iMoon: building 3D models of the indoor environment to solve the problem of missing paths by integrating paths into a navigation mesh. The above studies focus on designing systems or developing applications for MVC; however, these works pay no attention to the user recruiting strategy. Hence, this paper can be regarded as an important module of the above systems and applications.

B. Recruitment Strategy in MCS

He *et al.* [23] propose the user recruitment protocol for vehicle-based crowdsourcing by predicting the trace of the vehicle. To maximize task coverage, Wang *et al.* [24] propose two algorithms to select some seeds in social networks. Wang *et al.* [25] reformulate the multitask assignment problem to minimize sensing quality thresholds, with the purpose of assigning a suitable set of tasks to each worker. Yu *et al.* [26] propose to utilize a location-based social network to build a framework for improving marketing effectiveness by carefully selecting users to serve offline events. Li *et al.* [27] propose a recommender system that selects a subset of participants to maximize the utility of the platform for effectively sensing the data. Pouryazdan *et al.* [28] provide a framework based on game theory to ensure truthfulness for a mobile crowdsensing system in the user recruitment stage. Wang *et al.* [29] provide a comprehensive review for diversifying the problem formulation and allocation algorithms together with future research opportunities. Wang *et al.* [30] propose a mobile crowdsensing task assignment scheme by allocating tasks that consider users' moving regularity. Liu *et al.* [31] propose a multitask allocation framework in mobile crowdsensing with biobjective optimization goals, named TaskMe, which is based on minimum cost maximum flow theory. Wang *et al.* [32] propose PSTasker, an MCS platform that aims to maximize the overall system utility by suitably allocating the tasks to users. The above works focus on proposing a user recruitment strategy to efficiently finish the mobile crowdsensing task. However, the above works are unable to address coverage and redundancy problems of data collecting in MVC; hence, they could not be directly used in the recruiting system for MVC.

III. User Recruitment System

A. Task Publishing Model

We design a generic task publishing model, which requires the task requester to publish the following data type:

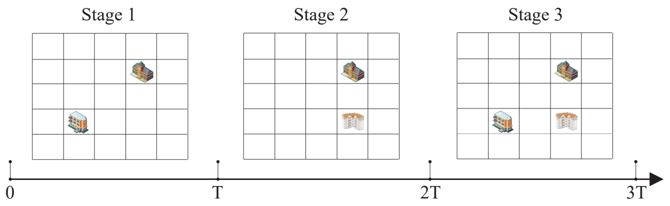


Fig. 2. The illustration of the multistage task publishing process for mobile crowdsensing.

$\langle T\_list, T\rangle$, where T_list means the target list consisting of the locations of targets (buildings or scenic spots) such that each target is regarded as a roundness in the map, and the target's information includes $\langle longitude, latitude$ and $diameter\rangle$. T means the deadline for the sensing task. Actually, the above task-publishing process is repeated serially as shown in Fig. 2; then, the multistage task publishing process is formulated. At the beginning of each stage, the task requester publishes the locations of the targets and the importance degrees of the targets. If some targets are not well constructed in the previous stage, then their importance degrees would be higher in this stage. The importance degree is quantified as the weight of the target, which will also influence the user recruitment. It is not difficult to realize that the different stages are independent of each other; hence, we use the single-stage recruiting problem as an example in this paper. Furthermore, the duration of a stage is set as one hour, which is not a long time; thus, we recruit the users at the beginning of the stage and will not adjust the recruiting before the stage deadline.

At the beginning of the stage, a requester publishes a sensing task to make reconstructions for a set of buildings. Then, the positions and deadline are also published, and we can know the longitude and latitude of the buildings' geometric centers and the diameter of the building. The detailed positions of the buildings can be further described. Moreover, the deadline to upload the photos of these buildings is also provided.

For each task, we have n users who are ready to be recruited: $U = \{u_1, u_2, \cdots, u_n\}$. Here, u_i refers to the user ID for taking the photos. According to the users' traces, we can decide the minimum size for the sensing area, which is a minimized rectangular area to cover all the users' historical mobility traces and sensing targets. When a requester publishes a sensing task, we can extract the detailed information about the sensing targets (position, size, *etc.*).

B. Map Gridding Model

Definition 1. *A* grid *is a square region characterized by* (i, j)*, which denotes the* i_{th} *row and* j_{th} *column grid in the map. Each grid has an equal size. A gridded map is a collection of disjoint grids that collectively cover the mobility area of a GPS dataset.*

It is not difficult to realize that a region on the Earth bounded by longitudinal and latitudinal lines is not a square. However, it is easy for us to map the user's longitudinal and latitudinal traces into a square area in a plane region, especially for a small area [33].

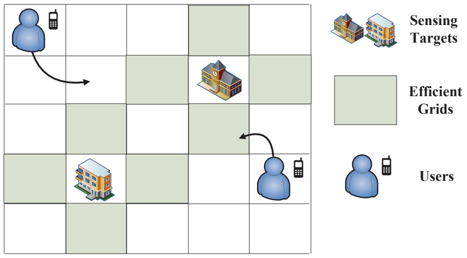


Fig. 3. The illustration of the gridded map for mobile crowdsensing.

After mapping each user's trace to a gridded map, we convert the trace to a string of grids. At any given time, the state of the user can be recorded as the grid ID. Generally, the more grids there are, the more overhead for maintaining the user's states but also the higher prediction precision of user mobility. Considering that the grid division is uniform, we decide the gridded map based on sensing targets' locations and the requirement of the task requester.

According to the targets' locations and sizes, we can decide the minimum size of the grid. Similarly, according to the shooting distance requirement of the target, we can decide the maximum size of the grid. Then, we partition the entire map area into grids by the following rules:

- Each grid contains and totally covers up to only one sensing target.
- The grids are equal size, and there is no overlap among them.
- The side length of the grid must be smaller than the required shooting distance.

Through the above rules, the map is gridded into $a \times b$ grids. The side length of a grid is larger than the maximum diameter of the target and smaller than the required maximum shooting distance. If a target covers multiple grids, without loss of generality, the target is assigned to the grid that covers its geometric center. Finally, we can achieve a gridded map as shown in Fig. 3, where a set of users move around the grids, and some grids are occupied by the sensing targets. It is worth noting that in different scenarios, the areas of targets may change, although we can always find a suitable gridding result through the above rules. Then, the north, south, east and west grids of the sensing target are defined as the efficient grids.

Definition 2. *An* efficient grid *is a grid where the user can take an efficient photo for a sensing target.*

The technology that reconstructs a target through the photos taken from multiple angles [34] has considerably matured. However, the reconstruction procedure is not the main part of this paper. The most important concept is that more photos lead to a higher modeling quality [35]. Hence, we use the number of nonredundant efficient photos to measure the performance of finishing the task. Therefore, we just propose a suitable recruitment strategy for maximizing the number of efficient photos that can be used to serve the target reconstructions. In this paper, according to the gridded map, we use the photos taken from four angles (north, south, east, and west) to perform the target reconstruction (as shown in Fig. 4). There have been many software offerings (Photo scan)

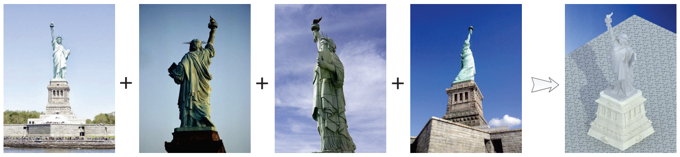


Fig. 4. An example of performing target reconstruction through the four photos of different angles (north, south, east and west).

or online applications (3D cloud) that can finish the above works. Meanwhile, the modeling procedure can be performed on a cloud server when the users' photos are successfully uploaded. Moreover, if we want to enhance the reconstruction quality, we could extend the work to a situation with more angles and modify the model of map gridding accordingly. In the following discussions, we explain the extension of this work to a multiangle situation through an easy improvement.

C. Utility Calculating Model

In the gridded map, the targets are located at the stationary grids, and the users move around the targets. Then, a user's trace could be regarded as the transition among the grids. When a user moves into an efficient grid, efficient photos for the corresponding targets can be taken. In an attempt to decide a user's utility for the specific task, we predict the expected number of efficient photos that a user could take before the deadline. Therefore, in this paper, we use the semi-Markov model to address the grid-based mobility prediction problem [8].

1) Semi-Markov Model: There are n users moving around the targets. The state of a user is defined as the grid ID it is currently in. Generally, the state set of user k is $L^k = \{(1,1),(1,2),(1,3),\cdots,(a,b)\}$, which represents the grid ID that user k is in now. $a \times b$ represents the total number of grids. The nth state of user k is recorded as L^k_n, which is the nth grid in the trace of user k. The entering time for user k to the nth grid is T^k_n.

Through the above transition, a user k's trace can be regarded as a time homogeneous semi-Markov process [36], [37] (L^k_n, T^k_n) because the probability of a user k moving from state L^k_n to state L^k_{n+1} has no relationship to the state L^k_{n-1}. Hence, L^k_n can be regarded as a standard discrete-time Markov chain. T^k_n represents the time point to move from L^k_n to L^k_{n+1}.

Moreover, we provide the following symbols: $M^k_n = T^k_{n+1} - T^k_n$, which represents the holding time in the grid. Obviously, M^k_n is independent and identically distributed (i.i.d.). Hence, the distribution for M^k_n is different from the geometric or exponential distribution. For example, each user may have its own expected holding time for the specific grid.

The main equation for the above semi-Markov model is shown in Eq. 1, where $F^{(x,y)}_{(i,j)}(k,t)$ is the probability for user k to move from grid (i,j) to grid (x,y) before time t. Obviously, in Eq. 1, L^k_{n+1} depends on L^k_n but is independent of L^k_{n-1}.

$$\begin{aligned} F^{(x,y)}_{(i,j)}(k,t) &= P(L^k_{n+1}{=}(x,y), M^k_n \leq t | L^k_0 \cdots L^k_n, T^k_0 \cdots T^k_n) \\ &= P(L^k_{n+1}{=}(x,y), M^k_n \leq t | L^k_n = (i,j)) \end{aligned} \quad (1)$$

Next, we consider P as the probability matrix for moving among the grids. Additionally, the holding states for a user satisfy a Markov chain. Then, the probability of moving from grid (i,j) to grid (x,y) is shown in Eq. 2, where $num_{(i,j)}(k)$ is defined as the number of moves out from grid (i,j) without considering the next grid, and where $num^{(x,y)}_{(i,j)}(k)$ is the number of moves from grid (i,j) to grid (x,y). It is worth noting that if (i,j) and (x,y) are not neighboring grids, then a user could not move directly from (i,j) to (x,y). In other words, for matrix P, only when the grids are neighbors in the gridded map may a user have a value to move between them. Hence, when $|i-x| > 1$ or $|j-y| > 1$, the value in P is 0.

$$P^{(x,y)}_{(i,j)}(k) = \begin{cases} P(L^k_{n+1}{=}(x,y) | L^k_n{=}(i,j)) \\ {=}num^{(x,y)}_{(i,j)}(k)/num_{(i,j)}(k), \\ \text{if } |\text{i} - \text{x}| \leq 1 \ \&\& \ |\text{j} - \text{y}| \leq 1 \\ \\ 0, else \end{cases} \quad (2)$$

The probability for user k to move from grid (i,j) to grid (x,y) before time t is symbolized as $T^{(x,y)}_{(i,j)}(k,t)$, which is shown in Eq. 3.

$$\begin{aligned} T^{(x,y)}_{(i,j)}(k,t) &= P(M^k_n \leq t | L^k_n = (i,j), L^k_{n+1} = (x,y)) \\ &= \sum_{c=1}^{t} P(M^k_n{=}c | L^k_n{=}(i,j), L^k_{n+1}{=}(x,y))) \end{aligned} \quad (3)$$

Then, we can calculate the probability $T_{(i,j)}(k,t)$ that user k will leave the grid (i,j) no later than time t as follows:

$$T_{(i,j)}(k,t) {=} P(M^k_n \leq t | L^k_n{=}(i,j)) {=} \sum_{\forall (x,y) \neq (i,j)} F^{(x,y)}_{(i,j)}(k,t). \quad (4)$$

As previously described, the state holding time is defined as M^k_n. Obviously, $T_{(i,j)}(k,t)$ also indicates the distribution of the grid-holding time in (i,j) for user k, regardless of the next grid or state.

According to Eqs. 1-3, we can derive the time-homogeneous semi-Markov kernel part $F^{(x,y)}_{(i,j)}(k,t)$, which is shown as Eq. 5.

$$\begin{aligned} F^{(x,y)}_{(i,j)}(k,t) &= P(L^k_{n+1}{=}(x,y), M^k_n \leq t | L^k_0 \cdots L^k_n, T^k_0 \cdots T^k_n) \\ &= P(M^k_n \leq t | L^k_n{=}(i,j), L^k_{n+1}{=}(x,y)) \\ &\quad P(L^k_{n+1}{=}(x,y) | L^k_n{=}(i,j)) \\ &= T^{(x,y)}_{(i,j)}(k,t) P^{(x,y)}_{(i,j)}(k) \end{aligned} \quad (5)$$

2) User Utility: Another main equation for the semi-Markov model is $R^{(x,y)}_{(i,j)}(k,t)$, which represents the probability that user k is now in grid (i,j), and after time t, the user k could be in grid (x,y). Obviously, $R^{(x,y)}_{(i,j)}(k,t)$ and the previous $F^{(x,y)}_{(i,j)}(k,t)$ are two different concepts, where $F^{(x,y)}_{(i,j)}(k,t)$ means the probability that user k's current grid is (i,j) and its next grid is (x,y), with the moving time from (i,j) to (x,y) less than t. However, $R^{(x,y)}_{(i,j)}(k,t)$ represents the probability that the user k's current grid is (i,j) and will be in (x,y) after time t. In other words, the moving process from (i,j) to (x,y) may pass through other grids. In this manner,

$R_{(i,j)}^{(x,y)}(k,t)$ can be used to predict the user k's location after a given time t once its current grid is known.

It is not difficult to realize that $\sum\limits_{\forall(x,y)} R_{(i,j)}^{(x,y)}(k,t) = 1$. For the start time $t = 0$ without any moving out, if $(x,y) \neq (i,j)$, then $R_{(i,j)}^{(x,y)}(k,t) = 0$. Similarly, if $(x,y) = (i,j)$, then $R_{(i,j)}^{(x,y)}(k,t) = 1$.

Aiming at calculating $R_{(i,j)}^{(x,y)}(k,t)$, we start with an easy situation: user k has never left grid (i,j) before time t. Then, according to Eq. 4, the probability that M_n^k is larger than t is shown in Eq. 6:

$$P(M_n^k > t | L_n^k = (i,j)) = 1 - T_{(i,j)}(k,t) \tag{6}$$

Then, another situation is considered when user k has been in at least another grid between time 0 and t, and the first moving out occurs at time c, such that user k moves to (r,f). Then, we can calculate the probability as follows:

$$\begin{aligned} & P(L_t^k{=}(x,y)|L_0^k{=}(i,j) \text{ and at least one transition to } (r,f)) \\ & = \sum_{\forall(r,f)} \sum_{c=1}^{t} (F_{(i,j)}^{(r,f)}(k,c) {-} F_{(i,j)}^{(r,f)}(k,c{-}1)) R_{(r,f)}^{(x,y)}(k,t{-}c), \end{aligned} \tag{7}$$

where $(F_{(i,j)}^{(r,f)}(k,c){-}F_{(i,j)}^{(r,f)}(k,c{-}1))$ represents that the first moving out for user k occurs at time c to grid (r,f).

Taking the above two situations into consideration, we obtain $R_{(i,j)}^{(x,y)}(k,t)$ as follows:

$$R_{(i,j)}^{(x,y)}(k,t) {=} \begin{cases} \sum\limits_{\forall(r,f)} \sum\limits_{c=1}^{t} (F_{(i,j)}^{(r,f)}(k,c) - F_{(i,j)}^{(r,f)}(k,c-1)) \\ R_{(r,f)}^{(x,y)}(k,t-c), (x,y) {\neq} (i,j) \\ 1 - T_{(i,j)}(k,t) + \\ \sum\limits_{\forall(r,f) \neq (i,j)} \sum\limits_{c=1}^{t} (F_{(i,j)}^{(r,f)}(k,c) - F_{(i,j)}^{(r,f)}(k,c-1)) \\ R_{(r,f)}^{(x,y)}(k,t-c), (x,y) {=} (i,j) \end{cases} \tag{8}$$

By now, a user's probability to be in a grid (x,y) before the sensing task's deadline is achieved. For an efficient grid, the higher the user's probability is, the more contribution it can make to the sensing task. Then, a user's utility for a specific grid is shown as follows:

$$U^{(x,y)}(k) {=} 1 - \prod_{t=0}^{T} (1 - R_{(a,b)}^{(x,y)}(k,t)) \tag{9}$$

In this subsection, we attempt to calculate the user's utility for taking the efficient photos for the sensing targets. Hence, we should use the sum probability for arriving at the efficient grids of user k to measure the expected number of efficient photos that could be taken. There are many efficient grids in the map, and the expected number of efficient photos that a user could take is just the sum probability of arriving at the efficient grids. However, the following two cases need to be considered; the first case is that, as previously discussed, a target has an importance degree. For example, if a target is not well reconstructed in the previous stage, then in the following stage, it is important to recruit the users who could contribute more efficient photos for this target. The second case is that a grid may be efficient for two or more targets; then, the grid (x,y) has a weight $W_{(x,y)}$, which is the sum of the weights of the surrounding targets adjacent to (x,y) (as shown in Fig. 5). In Fig. 5, the weight of the dark grid is 3 because it is adjacent to the two targets, whose weights are 1 and 2. Through the above analysis, the user's utility can be calculated as follows:

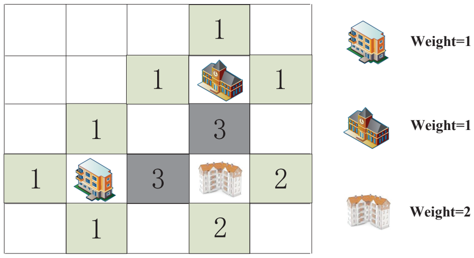


Fig. 5. The descriptions for the efficient grids with the weight. Each target has an importance degree, as previously discussed, and the degree is quantized as the weight of the target. Then, the initial weight of the grid around the target is just the weight of the target. However, if a grid is efficient for two targets, in other words, a user in this grid could take two efficient photos for the different targets, then the weight of this grid is the sum of the targets' weights.

$$U(k) {=} \sum_{\forall(x,y) \in E} W_{(x,y)} U^{(x,y)}(k) \tag{10}$$

D. User Recruitment Model

Now, the user's utility is determined. Based on the user's utility, we can propose a top-k user recruitment strategy, which recruits k users to collectively take the maximum number of efficient photos.

Given a user set S, for all the users $k \in S$, the probability that they could be in grid (x,y) before the deadline is shown as follows:

$$U^{(x,y)}(S) {=} 1 - \prod_{\forall k \in S} (1 - U^{(x,y)}(k)) \tag{11}$$

Then, the expected number of efficient photos that they could take can be achieved. It is not difficult to determine that for all the efficient grids set E, the utility of the user set S (i.e., the expected number of efficient photos that user set S could take) can be obtained as follows:

$$U_S {=} \sum_{\forall(x,y) \in E} W_{(x,y)} U^{(x,y)}(S) \tag{12}$$

1) Top-k User Recruitment: Before addressing the above user recruitment problem, we first try to prove that the top-k user recruitment problem is NP-hard, as shown in the following theorem [38].

Theorem 1. *The k nodes selection problem is NP-hard.*

Proof. First, an easier situation is considered in this paper: every user's utility for a specific grid is 1 or 0, which means that whether the user will be at a specific efficient grid before the deadline is a deterministic event, not a probabilistic event. Actually, the purpose of this k-user recruitment problem is to recruit the k number of users (user set S) from all the users to arrive at as many efficient grids as possible. This problem

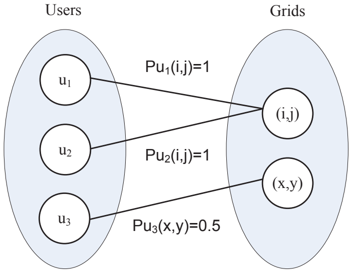


Fig. 6. The descriptions of two different user recruitment strategies. A top-2 user recruitment strategy is considered. If we just recruit the users with the highest utility for the two grids, (i, j) and (x, y), then users 1 and 2 will be recruited. However, users 1 and 3 are obviously a better user set (total utility is 1.5) than users 1 and 2 (total utility is 1).

can be considered a k set cover problem, which is a well-known NP-hard problem: a task set κ is provided, a collection of subsets is $\{\kappa_i | 1 \leq i \leq n\}$, and thus, we seek to determine a k size of the subcollection of $\{\kappa_i | 1 \leq i \leq n\}$ that covers as many tasks as possible in κ. That is, the easier (compared with the original k-user recruitment problem) k recruiting problem is NP-hard. Obviously, the original k-user recruitment problem is at least NP-hard. □

As previously discussed, the top-k user recruitment problem is NP-hard. Therefore, in order to address the NP-hard problem, we attempt to use a greedy algorithm. One approach that occurred to us was that according to every user's utility, we choose the user with the highest utility, and the same actions are performed k times. However, the above recruitment strategy is obviously not the optimal solution. For example, as shown in Fig. 6, users 1 and 2 could both arrive at grid (i, j) before the deadline; meanwhile, user 3 has a probability of 0.5 to arrive at grid (x, y). (i, j) and (x, y) are two efficient grids. Considering a top-2 user recruitment problem, if we recruit the two users with the highest utility, then users 1 and 2 would be recruited, and their total utility is 1 (1+0). However, a better strategy is to recruit users 1 and 3 because the utility of users 1 and 3 is 1.5 (1+0.5).

Aiming at selecting the top k users and achieving the maximum number of efficient photos, we propose the greedy heuristic strategy to recruit the best users set S. The purpose is to maximize U_S, which is calculated as Eq. 12. The detailed greedy algorithm is shown in Algorithm 1. The logic of Algorithm 1 is first recruiting the best user with the highest utility and, then, among the rest of the users, selecting the user who can maximize the value of U_S as the second recruiter to be added into S. The same actions are performed k times. As shown in Fig. 6, in this way, we select users 1 and 3 (or 2 and 3), rather than users 1 and 2 as the final recruiters.

2) Approximation Ratio: We have proposed a greedy algorithm to address the above NP-hard problem. Then, for tackling the complexity of the algorithm, the bound is derived in this subsection.

Theorem 2. *U_S is a submodular function, which means that for two arbitrary user sets S_1 and S_2, if $S_1 \subseteq S_2$, then $\forall u_k \notin S_2$, the submodular property holds, i.e., $U_{S_1 \cup \{u_k\}} - U_{S_1} \geq U_{S_2 \cup \{u_k\}} - U_{S_2}$.*

Algorithm 1 Greedy heuristic for recruitment of k users

```
Input:
    Number of users: $K$
    Recruited users: $S$
    $S$' total utility: $U_S$
Output:
    Top-$k$ users set: $S$
$S \longleftarrow \varnothing$; $U_S = 0$
for $i$ =1 to $k$ do
    $h \longleftarrow \arg\max_{h \in K \setminus S} U_{S \cup h}$
    $S = S \cup h$; update $U_S$
return $S$
```

Proof. We first prove that when $|S_2| - |S_1| = 1$, $U_{S_1 \cup \{u_k\}} - U_{S_1} \geq U_{S_2 \cup \{u_k\}} - U_{S_2}$. Then, we extend it to the general case, where $|S_2| - |S_1| = \omega > 1$.

First, without loss of generality, we let $S_2 \backslash S_1 = \{u_h\}$ according to $S_1 \subseteq S_2$, then $|S_2| - |S_1| = 1$. To prove the submodular property of U_S, we consider the arriving probability to one efficient grid (x, y), which can be divided into the following three cases [38]:

Case 1: u_k has no chance to be in (x, y). For this case, $U^{(x,y)}(k) = 0$. Therefore, we have $U_{S_1 \cup \{u_k\}} = U_{S_1}$ and $U_{S_2 \cup \{u_k\}} = U_{S_2}$. As a result, $U_{S_1 \cup \{u_k\}} - U_{S_1} = U_{S_2 \cup \{u_k\}} - U_{S_2} = 0$.

Case 2: u_k has an arriving probability to (x, y), but u_h has no chance to be in (x, y). For this case, $U^{(x,y)}(h) = 0$. Then, $U_{S_2} = U_{S_1 \cup \{u_h\}} = U_{S_1}$, and $U_{S_2 \cup \{u_k\}} = U_{S_1 \cup \{u_k\} \cup \{u_h\}} = U_{S_1 \cup \{u_k\}}$. Consequently, we can obtain $U_{S_1 \cup \{u_k\}} - U_{S_1} = U_{S_2 \cup \{u_k\}} - U_{S_2}$.

Case 3: Both u_k and u_h have an arriving probability to (x, y). Then, for all the users in S_1, the total arriving probability to (x, y) is defined as $P_1^{(x,y)}$. Similarly, for S_2, the arriving probability to (x, y) is defined as $P_2^{(x,y)}$. Obviously, $P_1^{(x,y)} \leq P_2^{(x,y)}$, and then, $U_{S_1 \cup \{u_k\}} - U_{S_1} = 1 - (1 - P_1^{(x,y)})(1 - U^{(x,y)}(k)) - P_1^{(x,y)}$. Similarly, $U_{S_2 \cup \{u_k\}} - U_{S_2} = 1 - (1 - P_2^{(x,y)})(1 - U^{(x,y)}(k)) - P_2^{(x,y)}$. Therefore, we have

$$\begin{aligned}
&(U_{S_2 \cup \{u_k\}} - U_{S_2}) - (U_{S_1 \cup \{u_k\}} - U_{S_1}) \\
&= (1 - (1 - P_2^{(x,y)})(1 - U^{(x,y)}(k)) - P_2^{(x,y)}) \\
&\quad - (1 - (1 - P_1^{(x,y)})(1 - U^{(x,y)}) - P_1^{(x,y)}) \\
&= (P_1^{(x,y)} - P_2^{(x,y)})U^{(x,y)}(k) \leq 0
\end{aligned} \tag{13}$$

Therefore, $U_{S_1 \cup \{u_k\}} - U_{S_1} \geq U_{S_2 \cup \{u_k\}} - U_{S_2}$.

In conclusion, $U_{S_1 \cup \{u_k\}} - U_{S_1} \geq U_{S_2 \cup \{u_k\}} - U_{S_2}$ holds for $\forall (x, y) \in E$ in all cases. Now, we consider the case of $|S_2| - |S_1| = \omega \geq 1$. Without loss of generality, we assume that $S_2 \backslash S_1 = \{u_h, u_{h+1}, \cdots, u_{h+\omega-1}\}$. Then, we have $U_{S_1 \cup \{u_k\}} - U_{S_1} \geq U_{S_1 \cup \{u_k\} \cup \{u_h\}} - U_{S_1 \cup \{u_h\}} \geq U_{S_1 \cup \{u_k\} \cup \{u_h\} \cup \{u_{h+1}\}} - U_{S_1 \cup \{u_h\} \cup \{u_{h+1}\}} \geq \cdots \geq U_{S_1 \cup \{u_k\} \cup \{u_h\} \cup \cdots \cup \{u_{h+\omega-1}\}} - U_{S_1 \cup \{u_h\} \cup \cdots \cup \{u_{h+\omega-1}\}} = U_{S_2 \cup \{u_k\}} - U_{S_2}$

Therefore, U_S is a submodular function. Theorem 2 is proven. □

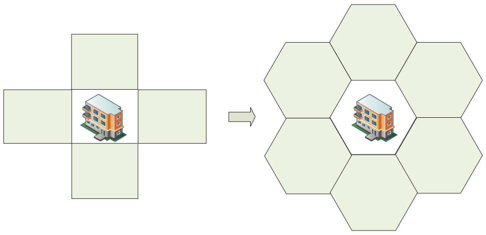


Fig. 7. An example of the irregularly gridded map.

Theorem 3. *For a nonnegative, monotone submodular function f, let S be a set of size k obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Let S^* be a set that maximizes the value of f over all k-element sets. Then, $f(S) \geq (1 - 1/e) \cdot f(S^*)$; in other words, S provides a $(1 - 1/e)$- approximation.*

Proof. Submodular functions have a very nice tractability property [39], which is relevant to us in the following way: we have a function f that is submodular, which takes only nonnegative values and is monotonic such that adding an element e to the set cannot cause f to decrease: $f(S \cup \{e\}) \geq f(S)$ for all elements e and sets S. We wish to find a k-element set S for which $f(S)$ is maximized. This is an NP-hard optimization problem (it can be shown to contain the set covering problem as a simplified case). However, a result from Nemhauser, Wolsey, and Fisher [40] shows that the following greedy hill climbing algorithm approximates the optimum to within a factor of $1-1/e$ (where e is the base of the natural logarithm): start with the empty set, and repeatedly add an element that yields the maximum marginal gain. □

E. Discussion

The proposed system model in this paper focuses on a gridded map as shown in Fig. 5. Actually, this is not a necessary condition; the system design in this paper is just a special case, undertaken to simplify the calculation. In other words, even if the map is not regularly gridded and the efficient grids are not located at the four angles (an example of six angles is shown in Fig. 7), the solution proposed in this paper could still be changed to satisfy the new problem. It is not difficult to realize that the grids in the map could be any shape, and we could still record the parameters mentioned in the utility calculating model. We use the grid ID to distinguish the different areas rather than the row and column numbers. For example, we could also record the grid holding time, the probability of moving out and so on. Then, based on the efficient grids, the user's utility is also achieved. Hence, the system model proposed in this paper could be easily extended to a new map area with an irregular map division.

IV. PERFORMANCE EVALUATION

A. The Traces Used and Settings

We adopt three real-world traces, *roma/taxi trace set* [41], *epfl trace set* [42], and *geolife trace set* [43], [8] to test the performances of the recruitment system. The *roma/taxi trace set* includes 320 taxi drivers that work in the center of Rome, Italy. The traces record the positions of drivers. Each taxi driver has a tablet that periodically retrieves a GPS position and sends it to a central server. The *epfl trace set* contains mobility traces of taxi cabs in San Francisco, CA, USA. It contains GPS coordinates of approximately 500 taxis collected over 30 days in the San Francisco Bay area. The *geolife trace set* contains 17,621 trajectories with a total distance of approximately 1.2 million kilometers and a total duration of approximately 48,000 hours. These trajectories are recorded by different GPS loggers and phones.

TABLE I
SIMULATION PARAMETERS

Parameter	**Traces**		
	roma/taxi	**epfl**	**geolife**
Simulation Time	650,660,· · · ,740,750		
Task Deadline	800		
Time Unit (s)	15	30	5
Grid Number	15×13	12×14	10×16
User Number	49	57	59
Target Number	7	7	7
Longitude	41.83∼41.96	37.72∼37.81	39.83∼39.97
Latitude	12.42∼12.56	-122.51∼-122.39	116.12∼116.52

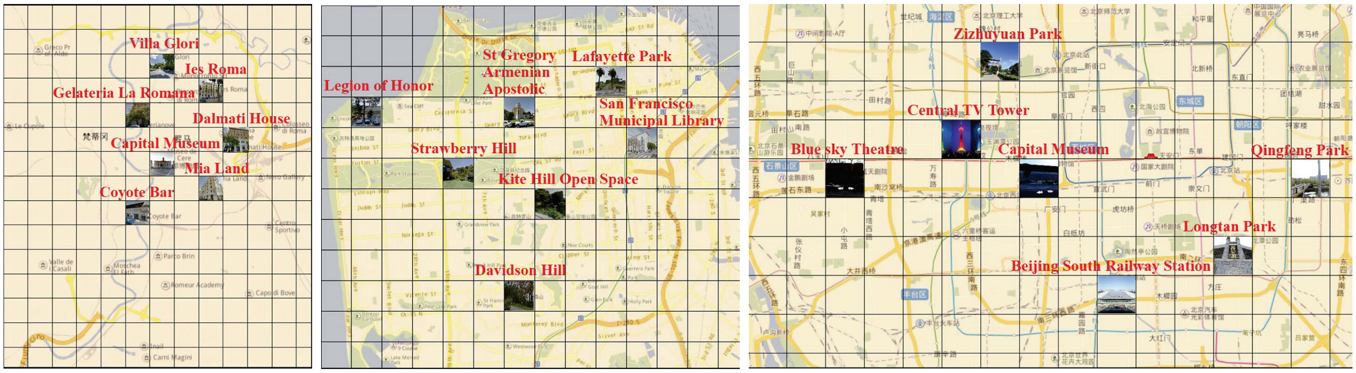
We first address these datasets by filtering out some abnormal user traces (discontinuous records or remote areas). According to the addressed users' traces, we achieve the map area for the users to move around. Then, we put the traces into the Baidu map according to the GPS records. Because the three traces are all in the city area, we randomly select some famous buildings or travel spots as the sensing targets. Based on the map gridding model, we decide the gridded map and the sensing targets (as shown in Fig. 8). The detailed simulation parameters in this network environment are listed in Table I. In particular, simulation time means the number of time slices, which is the period to collect data. The collecting period is 15 s for the *roma/taxi* dataset, 30 s for the *efpl* dataset and 5 s for the *geolife* dataset.

B. Algorithms and Performance Comparison

To demonstrate the performance of the designed user recruitment system in mobile crowdsensing, we have performed some simulations to evaluate two goals: (1) accuracy in the user's utility and (2) efficiency of the top-k users.

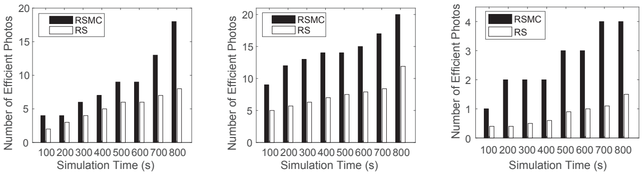
For the first part, to test whether the utility calculation of RSMC is accurate, we compare two recruitment strategies: RSMC and RS. RSMC is proposed in this paper for recruiting one user of the highest utility to take photos for the sensing targets. Therefore, at the system start time, we estimate the expected number of efficient photos that every user could take before the deadline and recruit the user with the highest number of efficient photos. RS (random selection) randomly recruits a user at the system start time.

For the second part, we attempt to test whether RSMC can achieve the highest sensing efficiency compared to the other two recruitment strategies: RSMC-H and RS. In this part, we focus on the top-k user recruitment problem. RSMC uses the system proposed in this paper to recruit the optimal k users (highest U_S) to take photos for the sensing targets. RSMC-H recruits k users that each have the highest utility; in



(a) roma/taxi (b) epfl (c) geolife

Fig. 8. The gridded map and sensing targets in the Baidu map of the three real-world datasets.



(a) *roma/taxi trace set* (b) *epfl trace set* (c) *geolife trace set*

Fig. 9. The performance comparison between RSMC and RS.

other words, RSMC-H does not consider the mutual influence among users' utilities. However, RS randomly recruits k users at the system start time.

While a range of data is gathered from the simulations, we take the following three main performance metrics into consideration:

(1) Number of efficient photos, which is the total number of efficient photos (without redundancy: taken in the same grid for the same target) that the recruiters could take before the deadline.

(2) Average delay, which is the average time needed for taking the efficient photos from the start time.

(3) Calculation time, which is the time needed for calculation of one layer of the R matrix, which is shown in Eq. 8.

Here, we emphasize that using the number of efficient photos instead of the number of completed tasks to measure the performance is implemented because the four angles are not necessary (it is just an example to show the modeling procedure). Therefore, it is difficult for us to quantify the number of completed tasks. To solve this problem, we use the number of efficient photos to measure the completion level of the sensing task.

C. Calculation Optimization

In terms of the above simulation, the main problem is the calculation speed because the calculation process consumes vast memory space. It is not difficult to realize that we need to calculate four kinds of matrices: P, T, F and R. Furthermore, each kind of matrix consists of many layers that are determined by the task deadline, and each layer is a matrix, including the square of the grid number. The grid number is the row number multiplied by the column number. For example, in the roma/taxi trace, for a P matrix, it is composed of 800 matrices, and the size of each matrix is 195×195. Hence, it occupies considerable memory space.

TABLE II
CALCULATION TIME FOR ONE LAYER OF THE MATRIX

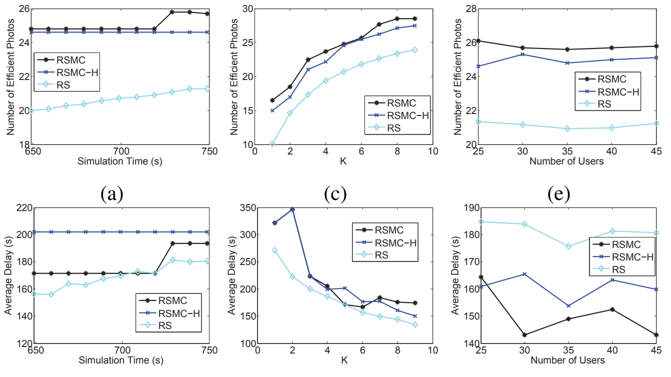
Matrix	Without Optimization	With Optimization
P	0.626 s	0.006 s
T	0.2547 s	0.0048 s
F	0.1085 s	0.0036 s
R	5.1 h	1 min

To optimize the calculation process, we first put the matrices on the disk, and the later calculation process would locate the needed item on the disk. Furthermore, for each user's four matrices, we move the data to the memory to accelerate the calculation speed because the different users' calculations can be parallel-processed, while for each user, its four matrices cannot be parallel-processed. Second, we further change to a three-tuple notation $\langle$*row number, column number, value*$\rangle$ to record the nonzero item in the original matrix. This is because the P, T, and F matrices are sparse, and using the three-tuple notation $\langle$*row number, column number, value*$\rangle$ to record the data can further improve the calculation speed.

Through the above two methods, the original calculation process is optimized, and we test the time for calculating one layer of the matrix, with the simulation results shown in Table II.

D. Simulation Results

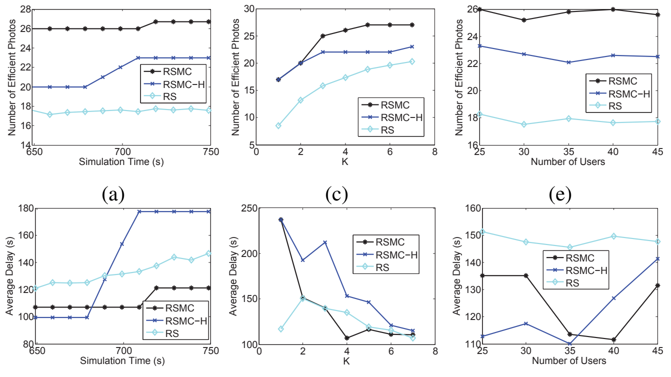
1) Accuracy of Utility: In the first part of the simulation results, we attempt to prove the accuracy of utility proposed in this paper. In other words, we attempt to test whether the user with the highest utility can actually take the largest number of efficient photos. To this end, we evaluate the performances of RSMC and RS, and we also conduct three groups of simulations using the *roma/taxi*, *epfl*, and *geolife* traces. At the beginning of the system time, RSMC recruits the user with the highest utility before the sensing deadline, and when the user enters an efficient grid, an efficient photo can be taken. At the end of deadline, the number of the efficient photos is regarded as the accuracy of utility. A higher number of efficient photos means a more accurate utility calculation. However, RS randomly selects a user as the recruiter. The same performance is tested in the three real-world traces. RSMC and RS are



(a) (c) (e)

(b) (d) (f)

Fig. 10. Performance comparisons on the *roma/taxi trace set*: number of efficient photos & average delay.



(a) (c) (e)

(b) (d) (f)

Fig. 11. Performance comparisons on the *epfl trace set*: number of efficient photos & average delay.

compared in this paper, and the number of efficient photos as a function of the simulation time is shown in Fig. 9 for the *roma/taxi*, *epfl*, and *geolife* traces.

As shown in Fig. 9-(a), in the *roma/taxi* real-world trace, along with the simulation time from 100 to 800, we test the number of efficient photos taken by RSMC and RS. It is not difficult to observe that the number of efficient photos taken by RSMC is always higher than that of RS. This is not difficult for us to understand because RSMC always recruits the user with the highest utility.

Moreover, there is an upward trend in the number of efficient photos along with the increase in the simulation time for RSMC and RS. This upward trend is mainly because a longer time leads to a higher probability for a user to pass an efficient grid; hence, the number of efficient photos increases. More importantly, along with the increase in the simulation time, the difference in terms of the number of photos between RSMC and RS also increases. The above phenomenon also proves that RSMC can always achieve a higher accuracy in terms of utility.

In Fig. 9-(b), we test the number of efficient photos taken by RSMC and RS along with the simulation time from 100 to 800 in the *epfl* real-world trace. It is not difficult to observe that RSMC always achieves a larger number of efficient photos than RS, which proves that the utility calculation proposed in this paper can recruit better users than randomly selecting users. Similarly, the difference in terms of the number of photos between RSMC and RS also increases with the increase in simulation time.

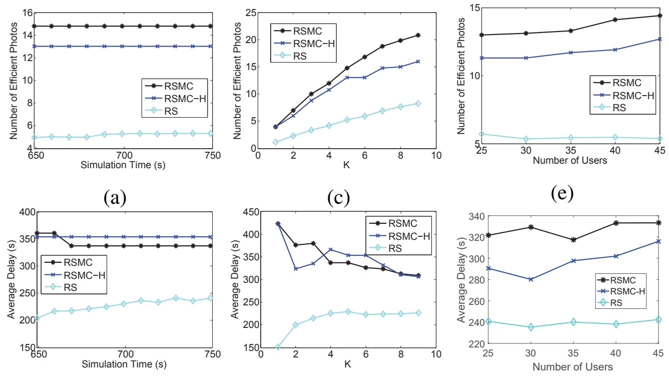
As shown in Fig. 9-(c), in the *geolife* real-world trace, we also test the number of efficient photos taken by RSMC and RS along with the simulation time from 100 to 800. The results show that in terms of the efficient photos performance, RSMC always achieves a higher efficiency than that of RS. This finding proves that the utility calculation proposed in this paper is suitable to recruit an efficient user. Moreover, because the efficient photos taken by the recruiters will increase along with the increase in simulation time, there is an upward trend of the number of efficient photos as the simulation time increases for both RSMC and RS. Similar to the previous simulations, the difference in terms of the number of photos between RSMC and RS also increases along with the increase in simulation time. In conclusion, in the *roma/taxi*, *epfl*, and *geolife* traces, RSMC always achieves a better performance for the number of efficient photos than that of RS.

2) Top-K Users Recruitment Efficiency: In this section, we focus on testing the performances of the top-k users recruitment strategy. In other words, we attempt to use the system proposed in this paper to recruit a set of the number k users to finish the crowdsensing task. We focus on the following two performances: number of efficient photos taken by the top k recruiters and the average delay for taking the efficient photos. To evaluate the performance of RSMC, we design two other recruitment strategies, RSMC-H and RS, which have been proposed in the previous subsection. We test the number of efficient photos and average delay, along with changing the simulation time, the value k, and the number of users in three real-world datasets. The simulation results are shown in Figs. 10-12.

As shown in Fig. 10, we first test the performances in the *roma/taxi* trace, and the performance ranking of the number of efficient photos is RSMC>RSMC-H>RS, which is reasonable and can match the theoretical result. Three conditions (simulation time, value of k and the number of users) are also taken into consideration to influence the final performances.

First, when the value of k is set to 5, the total number of users is 49, and the simulation time is changing, the number of efficient photos appears to be an upward trend for the RSMC recruitment strategy. It is easy for us to understand because a longer time leads to a larger number of sensing photos. For the RSMC-H, we recruit the users with the highest utility, and the simulation time's changing scope is not very large; thus, the users with the highest utility may be the same user set. Consequently, the RSMC-H may recruit the fixed users along with different simulation time situations. The above phenomenon appears in Fig. 10, where the number of efficient photos taken by RSMC-H remains unchanged in the different simulation times. However, the number of efficient photos taken by RS appears to be an upward trend along with the increase in the simulation time.

Second, the simulation time is set to 700, and the total number of users is 49. Along with the increase in the value k, we also test the number of efficient photos, as shown in Fig. 10. Obviously, there is also an upward trend for the RSMC recruitment strategy. This is correct because a larger number of recruiters leads to a larger number of efficient sensing



(a) (c) (e)

(b) (d) (f)

Fig. 12. Performance comparisons on the *geolife trace set*: number of efficient photos & average delay.

photos. It is not difficult to realize that the three recruitment strategies all appear to be an upward trend along with the increase in the value k. However, RSMC always achieves the highest performance compared with RSMC-H and RS. More importantly, the performance ranking of the number of efficient photos is also RSMC>RSMC-H>RS, which proves that RSMC recruits the optimal k users to finish the sensing task. This finding also matches our theoretical results.

Finally, the simulation time is set to 700, the value of k is set to 5, and along with the increase in the total user number, we test the number of efficient photos and the average delay. The simulation results show that RSMC still achieves the highest number of efficient photos along with the change of the total user number. In conclusion, in the *roma/taxi* real-world trace, the number of efficient photos taken by RSMC is larger than the other two strategies, RSMC-H and RS, along with the increases in the simulation time, the value of k and the number of users.

Then, in Fig. 11, the two previous performances in the *epfl* trace are tested. The simulation results show that the performance ranking of the number of efficient photos is RSMC>RSMC-H>RS, along with the change of simulation time, value of k and the number of users. The simulation results match the theoretical derivations.

First, the value of k is set to 4, the total number of users is 57, and along with the change of simulation time, the number of efficient photos appears to be an upward trend for the RSMC recruitment strategy. This result is because a longer time leads to a larger number of sensing photos. A similar shape also appears for RSMC-H. More importantly, RSMC always achieves the largest number of efficient photos compared with RSMC-H and RS.

Second, the simulation time is set to 700, and the total number of users is 57. Along with the increase in the value k, we also test the number of efficient photos, as shown in Fig. 10. There is also an upward trend for the RSMC recruitment strategy. More importantly, the performance ranking of the number of efficient photos is still RSMC>RSMC-H>RS, which proves that RSMC recruits a better group of k users to finish the sensing task. This is also our theoretical result.

Third, the simulation time is set to 700, and the value of k is set to 4. Along with the increase in the total user number, we test the number of efficient photos and average delay. The simulation results show that RSMC still achieves the highest number of efficient photos along with the change of the total user number. In conclusion, in the *epfl* real-world trace, the number of efficient photos taken by RSMC is larger than that of the other two strategies, RSMC-H and RS, along with the increases in the simulation time, the value of k and the number of users.

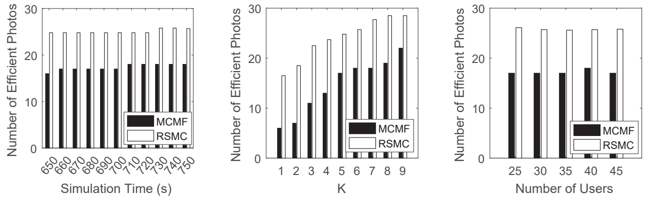


Fig. 13. Performance comparisons on the *roma/taxi trace set*.

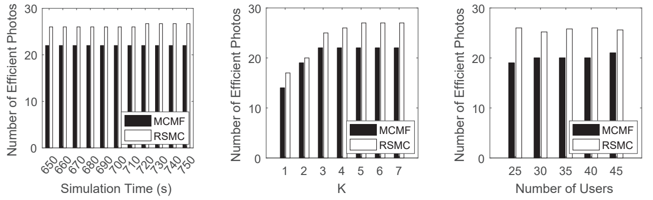


Fig. 14. Performance comparisons on the *epfl trace set*.

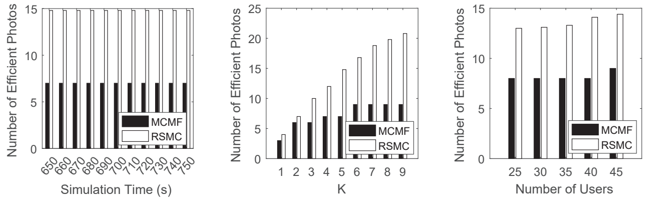


Fig. 15. Performance comparisons on the *geolife trace set*.

Finally, as shown in Fig. 12, we test the performances in the *geolife* trace. The performance ranking of the number of efficient photos is still RSMC>RSMC-H>RS, and the simulation sets and results are also similar to those of the previous simulations. Three conditions (simulation time, value of k and the number of users) are also taken into consideration to influence the final performances.

To further prove the efficiency of the proposed RSMC, in the previous simulation settings, we test the performance of the number of efficient photos compared with the work [31]. The simulation results are shown in Figs. 13-15, where RSMC achieves a better performance than MCMF. This result is because MCMF selects users according to the traveling cost of the TSP problem, and the purpose of this work is to minimize the cost for the users to finish the sensing task; therefore, it does not consider the coverage effects among users. However, the purpose of RSMC is to cover as many efficient areas as possible.

V. Conclusion

We have designed a user recruitment system for efficient photo collecting in mobile crowdsensing, which includes the following four submodels: task publishing, map gridding, utility calculating and user recruitment. First, we decide the sensing targets' locations through a task publishing model. Then, the map is gridded into subareas, and the efficient grids are decided. Furthermore, we use the semi-Markov model to predict the probability that a user enters an efficient grid before the deadline. Based on the targets' locations, the expected number of efficient photos that a user can take is decided, and the user's utility is also determined. Finally, we formulate the top-k user recruitment problem as an NP-hard problem, and the submodular method is used to solve the NP-hard problem. We conduct extensive simulations based on three widely used real-world traces: *roma/taxi*, *epfl*, and *geolife*. The results show that compared with other recruitment strategies, the recruiting system RSMC takes the largest number of efficient photos for the sensing task.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[2] Y. Gao, W. Dong, K. Guo, X. Liu, Y. Chen, X. Liu, J. Bu, and C. Chen, "Mosaic: A low-cost mobile sensing system for urban air quality monitoring," in *Proc. of IEEE INFOCOM 2016*.

[3] Y. Hu, G. Dai, J. Fan, Y. Wu, and H. Zhang, "Blueaer: A fine-grained urban pm2.5 3d monitoring system using mobile sensing," in *Proc. of IEEE INFOCOM 2016*.

[4] R. Du, C. Fischione, and M. Xiao, "Flowing with the water: On optimal monitoring of water distribution networks by mobile sensors," in *Proc. of IEEE INFOCOM 2016*.

[5] E. Koukoumidis, L.-S. Peh, and M. Martonosi, "Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proc. of ACM MobiSys 2011*.

[6] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A Survey of Mobile Phone Sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[7] J. Wan, J. Liu, Z. Shao, A. V. Vasilakos, M. Imran, and K. Zhou, "Mobile crowd sensing for traffic prediction in internet of vehicles," *Sensors*, vol. 16, no. 1, 2016.

[8] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–14, 2017.

[9] B. Guo, Q. Han, H. Chen, L. Shangguan, Z. Zhou, and Z. Yu, "The emergence of visual crowdsensing: Challenges and opportunities," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, pp. 1–18, 2017.

[10] Y. Wu, Y. Wang, W. Hu, and G. Cao, "Smartphoto: A resource-aware crowdsourcing approach for image sensing with smartphones," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1249–1263, 2016.

[11] S. Kim, C. Robson, T. Zimmerman, J. Pierce, and E. M. Haber, "Creek watch: pairing usefulness and usability for successful citizen science," in *Proc. of ACM CHI 2011*, pp. 2125–2134.

[12] M. Y. S. Uddin, H. Wang, F. Saremi, G.-J. Qi, T. Abdelzaher, and T. Huang, "Photonet: a similarity-aware picture delivery service for situation awareness," in *Proc. of IEEE RTSS 2011*, pp. 317–226.

[13] K.Tuite, N. Snavely, D.-y.Hsiao, N. Tabing, and Z. Popovic, "Photocity: training experts at large-scale image acquisition through a competitive game," in *Proc. of ACM CHI 2011*, pp. 1383–1392.

[14] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "Wreckwatch: Automatic traffic accident detection and notification with smartphones," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 285–303, 2011.

[15] B. Guo, H. Chen, Q. Han, Z. Yu, D. Zhang, and Y. Wang, "Worker-contributed data utility measurement for visual crowdsensing systems," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2379–2391, 2017.

[16] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "iCrowd: Near-Optimal Task Allocation for Piggyback Crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2010–2022, 2016.

[17] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Transactions on Mobile Computing*, DOI: 10.1109/TMC.2018.2829506, 2018.

[18] T. Li, H. Chang, M. Wang, B. Ni, and R. Hong, "Crowded scene analysis: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 367–386, 2015.

[19] H. Chen, B. Guo, Z. Yu, and Q. Han, "Toward real-time and cooperative mobile visual sensing and sharing," in *Proc. of IEEE INFOCOM 2016*, pp. 1–9.

[20] ——, "A generic framework for constraint-driven data selection in mobile crowd photographing," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 284–296, 2017.

[21] Q. Wang, B. Guo, L. Wang, T. Xin, H. Du, H. Chen, and Z. Yu, "Crowdwatch: Dynamic sidewalk obstacle detection using mobile crowd sensing," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–13, 2017.

[22] J. Dong, Y. Xiao, M. Noreikis, Z. Ou, and A. Yla-Jaaski, "imoon: Using smartphones for image-based indoor navigation," in *Proc. of ACM SenSys 2015*, pp. 85–97.

[23] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *Proc. of IEEE INFOCOM 2015*.

[24] J. Wang, F. Wang, Y. Wang, D. Zhang, L. Wang, and Z. Qiu, "Social-network-assisted worker recruitment in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–13, 2018.

[25] J. Wang, Y. Wang, D. Zhang, F. Wang, H. Xiong, C. Chen, Q. Lv, and Z. Qiu, "Multi-task allocation in mobile crowd sensing with individual task quality assurance," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2101–2113, 2018.

[26] Z. Yu, D. Zhang, Z. Yu, and D. Yang, "Participant selection for offline event marketing leveraging location-based social networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 6, pp. 853–864, 2015.

[27] W. Li, F. Li, K. Sharif, and Y. Wang, "When user interest meets data quality: A novel user filter scheme for mobile crowd sensing," in *Proc. of IEEE ICPADS 2017*, pp. 1–8.

[28] M. Pouryazdan, C. Fiandrino, B. Kantarci, T. Soyata, D. Kliazovich, and P. Bouvry, "Intelligent gaming for mobile crowdsensing participants to acquire trustworthy big data in the internet of things," *IEEE ACCESS*, vol. PP, no. 99, pp. 1–15, 2017.

[29] J. Wang, L. Wang, Y. Wang, D. Zhang, and L. Kong, "Task allocation in mobile crowd sensing: State-of-the-art and future opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3747–3757, 2018.

[30] L. Wang, Z. Yu, B. Guo, F. Yi, and F. Xiong, "Mobile crowd sensing task optimal allocation: A mobility pattern matching perspective," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 231–244, 2018.

[31] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "Taskme: Multi-task allocation in mobile crowd sensing," in *Proc. of ACM UbiComp 2016*.

[32] J. Wang, F. Wang, Y. Wang, D. Zhang, B. Y. Lim, and L. Wang, "Allocating heterogeneous tasks in participatory sensing with diverse participant-side factors," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–13, 2018.

[33] M. Lin, W.-J. Hsu, and Z. Q. Lee, "Predictability of individuals' mobility with high-resolution positioning data," in *Proc. of ACM Ubicomp 2012*, pp. 1–10.

[34] T. Schoeps, J. L. Schönberger, and S. G. et al, “A multi-view stereo benchmark with high-resolution images and multi-camera videos,” in *Proc. of CVPR 2017*.

[35] A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum, “Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks,” in *Proc. of IEEE CVPR 2017*.

[36] Q. Yuan, I. Cardei, and J. Wu, “An Efficient Prediction-Based Routing in Disruption-Tolerant Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 19–31, 2012.

[37] E. Wang, Y. Yang, and L. Li, “A Clustering Routing Method Based on Semi-Markov Process and Path-Finding Strategy in DTN,” *Chinese Journal of Computers*, vol. 38, no. 3, pp. 483–499, 2015.

[38] M. Xiao, J. Wu, H. Huang, L. Huang, and C. Hu, “Deadline-sensitive user recruitment for mobile crowdsensing with probabilistic collaboration,” in *Proc. of IEEE ICDCS 2016*.

[39] Y. Yang, Y. Xu, E. Wang, K. Lou, and D. Luan, “Exploring influence maximization in online and offline double-layer propagation scheme,” *Information Sciences*, vol. 450, no. 2018, pp. 182–199, 2018.

[40] G. Comuejols, M. Fisher, and G. Nemhauser, “Location of bank accounts to optimize float.” *Management Science*, vol. 1977, no. 23.

[41] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, “CRAWDAD dataset roma/taxi (v. 2014-07-17),” Downloaded from http://crawdad.org/roma/taxi/20140717, 2014.

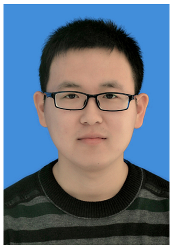
[42] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, “CRAWDAD dataset epfl/mobility (v. 2009-02-24),” Downloaded from http://crawdad.org/epfl/mobility/20090224, Feb. 2009.

[43] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma., “Mining interesting locations and travel sequences from gps trajectories,” in *Proc. of ACM WWW 2009*.

En Wang received his B.E. degree in software engineering from Jilin University, Changchun, in 2011, his M.E. degree in computer science and technology from Jilin University, Changchun, in 2013, and his Ph.D. in computer science and technology from Jilin University, Changchun, in 2016. He is currently an associate professor in the Department of Computer Science and Technology at Jilin University, Changchun. He is also a visiting scholar in the Department of Computer and Information Sciences at Temple University in Philadelphia. His current research focuses on the efficient utilization of network resources, scheduling and drop strategy in terms of buffer-management, energy-efficient communication between human-carried devices, and mobile crowdsensing.

Yongjian Yang received his B.E. degree in automatization from Jilin University of Technology, Changchun, Jilin, China, in 1983; his M.E. degree in computer communication from Beijing University of Post and Telecommunications, Beijing, China, in 1991; and his Ph.D. in software and theory of computer from Jilin University, Changchun, Jilin, China, in 2005. He is currently a professor and a PhD supervisor at Jilin University, Director of Key lab under the Ministry of Information Industry. His research interests include: network intelligence management, wireless mobile communication and services, and wireless mobile communication.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.



Kaihao Lou received his B.E.degree in Software Engineering from Jilin University, Changchun, Jilin,China, in 2017. He is currently a postgraduate student in Computer System Architecture from Jilin University, Changchun, Jilin, China. His current research focuses on Mobile Crowdsensing.

Dongming Luan received his B.E.degree in Software Engineering from Jilin University, Changchun, Jilin, China, in 2017. Now, he is a postgraduate student in College of Computer Science and Technology in Jilin University, Changchun, Jilin, China. His current research interest is Mobile Crowdsensing.

Hengzhi Wang graduated from Jilin University Software Engineering, bachelor’s degree in 2017. Now he is studying for a master’s degree at the College of Computer Science and Technology, Jilin University. His recent research are Mobile Crowdsensing and Network Economics.