Privacy-aware High-quality Map Generation with Participatory Sensing

Xi Chen, *Student Member, IEEE,* Xiaopei Wu, *Member, IEEE,* Xiangyang Li, *Fellow, IEEE,* Xiaoyu Ji, *Student Member, IEEE,* Yuan He, *Member, IEEE,* and Yunhao Liu, *Fellow, IEEE*

Abstract—Accurate maps are increasingly important with the growth of smart phones and the development of location-based services. Several crowdsourcing based map generation protocols that rely on users to provide their traces have been proposed. Being creative, however, those methods pose a significant threat to user privacy as the traces can easily imply user behavior patterns. On the flip side, crowdsourcing-based map generation method does need individual locations. To address the issue, we present a systematic participatory-sensing-based high-quality map generation scheme, PMG, that meets the privacy demand of individual users. To be specific, the individual users merely need to upload unorganized sparse location points to reduce the risk of exposing users' traces and utilize the $Crust$, a technique from computational geometry for curve reconstruction, to estimate the unobserved map as well as evaluate the degree of privacy leakage. Experiments show that our solution is able to generate high-quality maps for a real environment that is robust to noisy data. The difference between the ground-truth map and the produced map is less than $10m$, even when the collected locations are about $32m$ apart after clustering for the purpose of removing noise.

Index Terms—privacy protection, map generation, curve reconstruction, data suppression, participatory sensing

1 Introduction

During the last decade, portable devices have significant improvements in terms of computing performance, memory size and the number of embedded sensors(*e.g.*, GPS, accelerometer and gyroscope). These improvements allow the devices to be adopted in more scenarios such as navigation, location-based services and etc. [1]–[7]. Most of the applications jointly exploit the integrated maps and users' current location to provide various services. Hence, it is fundamental and indispensable to provide accurate and most-updated maps. Currently, digital maps based on the satellite images and street level information are widely used. But they cannot precisely reflect the most up-to-date ground information, especially in the developing countries, when cities are often under constructions and renovations, the integrated maps are likely to be far behind the current state.

To reflect the map dynamics accurately and effectively, several techniques have been proposed recently, among which participatory sensing attracts the most attention. Individual users contribute their trace information (with GPS data) to a central map generation server. While guaranteeing high quality of map information, the existing methods have various limitations, such as energy inefficiency and privacy leakage [36]. In this study, we design a privacy-aware map generation scheme, PMG. Unlike the existing methods [8]–[13], in our scheme, each user selectively chooses, reshuffles, and uploads a few locations from their traces, instead of the entire traces. After receiving those *unorganized* points from a group of users, the server generates the final map.

To provide high-quality map generation service, meanwhile preserving the privacy for each user, there are three major challenges we need to address: 1) quantifying the privacy leakage of data points provided by individual users; 2) generating theoretically-proven map using the reported unorganized points cloud; 3) designing map generation scheme that is robust to various discrepancies such as GPS error.

Directly reporting traces is not a good choice for protecting user's privacy. In PMG, we let each individual user select a subset of points from real traces, so that a user could protect his privacy from two aspects. The first is to break the temporal relationship among reported points. We let the user shuffle the points from his trace and then report the shuffled partial collection to the server, for obscuring the temporal relationship among original points. The second aspect is to limit the number of points reported in a region during a time-window. The challenge is to decide how many and which points a user has to select and report and we propose a mathematical formula to quantify the relationship between the reported locations and the degree of privacy leakage.

In the server, the fundamental task is to reconstruct the underlying map from a group of unorganized location points. Clearly, we cannot rely on the traditionally trace-based map generation method (*e.g.*, CrowdAtlas [2]) that sequentially connects the points according to the

- *Xi Chen, Xiaopei Wu, Yuan He, and Yunhao Liu are with School of Software and TNLIST, Tsinghua University, China.*
Email: {chenxi198909,wuxiaopei84}@gmail.com, {he,yunhao}@greenorbs.com
- *Xiangyang Li is with Department of Computer Science, Illinois Institute of Technology, Unite States, and with School of Software and TNLIST, Tsinghua University, China.*
Email: xli@cs.iit.edu
- *Xiaoyu Ji is with Department of Computer Science, HKUST, Hong Kong.*
Email:xji@cse.ust.hk

sampled time label, since two adjacent points may not be consecutive in any trace. Thus, under the privacy-preserving, it is not a trivial task to seek for an effective map generation algorithm with theoretical performance guarantee. We address the challenge of building a high-quality map from a set of unorganized points by using theoretically sound *curve reconstruction* techniques from computational geometry. When the sampling (a set of points reported by all users) reaches a certain threshold, the quality of the generated map is assured.

The third challenge is to design a map generation algorithm that is robust to noisy data. Typically, the GPS data has an error at least $10m$. The sparsity of the sampled locations, the small local feature size at some portions of the map, and the GPS error, will lead to inaccurate or even erroneous generated map. To overcome these problems, we apply a simple GPS data filtering procedure to remove all potential unreliable data. By requesting sufficiently dense samples and carefully clustering the reported points, this scheme can be robust to GPS errors.

There are also many subtle details that need to be carefully considered. For example, a critical component for the map generator here is to decide where to query the crowd for points that will produce the best possible map under certain resource constraints. We formulate our problem into the classical problem of location selection with the goal of maximizing the Lower Bound of Voronoi Angle (LBVA) criterion, meanwhile satisfying the minimal requirement of privacy protection. We show that such a problem is NP-hard and propose a simple heuristic with theoretically proven bound on the map quality that is within a constant factor of the optimum.

We extensively evaluate this design based on two real, high-resolution, city-scale GPS trace data. Our results show that the distance between the ground truth map and the map generated by our scheme PMG is less than $10m$. In our experiments, after the filter-out by each user for privacy-protection, the sampled points are about $7.5m$ apart on average. As these sampled points are inherently noisy due to GPS errors, we cluster them to produce "smoothed" samples for map generation. The smoothed sample points are actually about $32m$ apart on average, sufficient for producing accurate map.

The rest of this paper is organized as follows. In Section 2 we formally define the map generation problem with privacy-protection, review the background of curve reconstruction, and point out the challenges of applying such theory into our context. Detailed solutions are presented in Section 3. We present our evaluation results in Section 4, review the related work in Section 5 and conclude the paper in Section 6.

2 Problem Formulation and Background

2.1 Problem Formulation

We assume that our map generation service is composed of one central data processing server and a group of users spread over a geographic region. The server is in charge of collecting data (submitted voluntarily by these users or queried by the server) and producing a high-quality map from the set of collected locations. For the map generation, we do not assume that the server has a prior knowledge of the map, although such knowledge will significantly improve the performance of our method. A group of users travel in a geographic region and can collect a stream of GPS location trace using smartphones. Each user will provide some transformed data of the traces to the server for map construction.

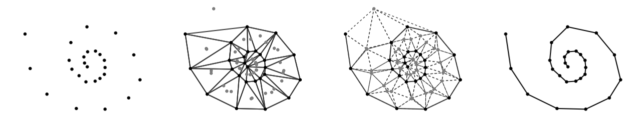
In this work, a map is mathematically defined as a geometric graph $G = (V, E)$ where V is the set of intersections in the map and E is the set of road segments connecting intersections. Consider one unobserved map $\mathcal{F}$. A simple naive solution of asking each user to report her/his traces directly will result in the disclosure of individual trace information which could be used to infer her/his identity and other behavior patterns, *e.g.*, where s/he lives, or even when s/he is away from home. To eliminate the possible risk of privacy exposure, one natural way is to let the user report fewer locations. However, this will inevitably affect the quality of map generation. To address the debacle between map quality and user's privacy, in this work we let each user upload a subset of GPS points (which are randomly shuffled to remove the temporal ordering of points in the trace) so as to minimize a certain measure of map generation errors. This approach can assure that some constraints on individual trace privacy exposure are satisfied. If not specified otherwise, throughout this paper the privacy we want to protect is the private trace/trajectory associated with each user.

More formally, consider m users and let $U_i (1 \leq i \leq m)$ be the set of collected GPS points by user i. To avoid potential privacy exposure, each user will carefully choose a subset of U_i, denoted as P_i, to report. Therefore, the optimal map generation problem (**P**) with privacy-preserving constraints is given as follows:

$$\textbf{(P)} \quad \cup_{i=1}^{m} P_i^* = \underset{\forall i, P_i \subseteq U_i}{\arg\min} \quad \mathrm{Err}\left(\mathcal{F}, \zeta(\cup_{i=1}^{m} P_i)\right)$$
$$\text{subject to} \quad \mathrm{PE}_i(P_i) \leq b_i, \ 1 \leq i \leq m,$$

where $\zeta(.)$ returns the estimated map given reported GPS point set from m users, Err() is a certain error function measuring the distance between the real map $\mathcal{F}$ and the estimated map $\zeta(.)$, $PE_i(.)$ is the privacy-exposure function that reflects the degree of privacy leak of user i and b_i is the corresponding privacy leakage constraint (called privacy budget sometimes) for publishing P_i. of revealing the individual private location profile.

A typically used error function is the mean-squared error, defined as $\|\mathcal{F} - \zeta(\cup_{i=1}^{m} P_i)\|_2$. To compute this metric, we need to know the original map $\mathcal{F}$ beforehand, which is often unavailable in practical setting. As an alternative, we will focus on the "quality" of the set of collected points. We later will show that, if the set of collected points meets certain sampling condition, the re-



(a) Initial sample S (b) The Voronoi diagram of S (c) $Del(S \cup V)$ (d) The *Crust* of S

Fig. 1: The process of *Crust*

constructed map will have a lower bound on the quality between the ground-truth map $\mathcal{F}$ and the reconstructed map. In fact, if we view the map as one polygonal curve in 2D plane and the point set $\cup_{i=1}^{m} P_i$ as samples with respective to that curve, the estimate function $\zeta(\cdot)$ will fall into the category of curve reconstruction [16] in computational geometry, which allows one to uniquely determine the original curve from a subset of samplings that satisfies some special conditions. It is therefore particularly attractive for our specific problem.

The degree of privacy disclosure highly depends on the data that users publish. The foremost task here is to quantify the privacy protection of the data submitted by each user. A simple measure would be the number of points reported by the user: uploading more points lead to worse privacy protection. So a user may put a limit on the number of points reported in a time-window (thus $\text{PE}_i(P_i)$ is simply the cardinality of set P_i). Obviously, reporting a large volume data in a small time-window is not preferred. Note that this naive cardinality constraint cannot quantify the privacy protection level in other metrics. For example, an adversary may still be able to infer some privacy information if P_i is a continuous subsegment in a trace. To quantify the ability of protecting the trace information of each user, we will introduce a novel privacy quantification function in Section 3.2. Intuitively, our privacy quantification assures that the adversary cannot recover the users' trace when certain conditions are met. Note that different privacy quantification functions could be integrated into our scheme, such as a function based on the Hidden Markov Model in [14] or the Bayes conditional risk in [15].

2.2 Curve Reconstruction

In this section we briefly review the background and techniques for curve reconstruction, a theoretical foundation of our map construction scheme.

Consider an unknown smooth curve $\mathcal{F}$. Given a set of unorganized points S sampled from $\mathcal{F}$, the curve reconstruction problem is to construct a graph containing exactly those edges that connect the adjacent points in $\mathcal{F}$.

Extensive effective approaches [17]–[19] have been proposed to find the solution of such a problem, among which *Crust* [16], one geometric graph, coincides with $\mathcal{F}$ if S satisfies some specific sampling conditions (more will be discussed below). We next will focus on *Crust* due to its simplicity, theoretical guarantees and good quality.

The *Crust* induced by S is a graph such that any edge is one element in $Del(S \cup Z)$, with only the points in S as its endpoints, where Z is the vertices of the Voronoi diagram induced by S and $Del(S \cup Z)$ returns the Delaunay triangulation of $S \cup Z$. Therefore, the *Crust* of S could be generated in three phases: (1) compute the Voronoi diagram of S; (2) calculate the Delaunay triangulation of $S \cup Z$, denoted by D; (3) remove all the edges in D unless both of their endpoints belong to S. Figure 1 illustrates the process of constructing the *Crust*.

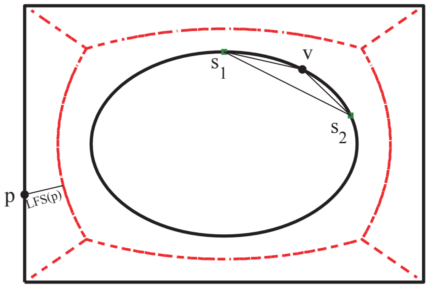


Fig. 2: Medial Axis (in red), LFS(p), and Voronoi Angle.

Due to the existence of advanced and elegant program for Delaunay triangulation [20] and Voronoi diagram [21], computing *Crust* of one given finite set S is easy to implement and scalable to the cardinality of S, with $O(n \log n)$ running time, where $n = |S|$. More importantly, the performance of *Crust* is theoretically guaranteed, *i.e.*, *Crust* provably solves the curve reconstruction problem under certain conditions. Before giving such specific result, we would like to cite four relative definitions in [16] at first.

Definition 1. *The* Medial Axis *of a curve $\mathcal{F}$ is closure of the set of points which have two or more closest points in $\mathcal{F}$.*

Definition 2. *The local feature size, LFS(p), of a point $p \in \mathcal{F}$ is the Euclidean distance from p to the closest point on the medial axis.*

Definition 3. *A curve $\mathcal{F}$ is γ-sampled by points set S if, $\forall p \in \mathcal{F}$, the closest sample $s \in S$ satisfying $\frac{D(p,s)}{LFS(p)} \le \gamma$, where $D(p, s)$ represents the distance between p and s.*

Definition 4. *A curve Voronoi disk is a maximal disk, empty of the samples S inside, centered at a point of the curve. A* curve Voronoi vertex *v is the center of a curve Voronoi disk. The angle $\angle s_1 v s_2$ is called Voronoi angle (e.g., $\angle s_1 v s_2$ in Fig. 2 if v is a curve Voronoi vertex and s_1, s_2 are on the boundary of curve Voronoi disk centered at v.*

These definitions are graphically shown in Fig. 2. The solid black curve represents the original smooth curve. And its corresponding medial axis is shown as the dashed red curve.

Armed with these definitions, we will give two useful theoretical analyses in [16], denoted as Lemma 1 and 2.

LEMMA 1. *Let S be a γ-sample from a smooth curve $\mathcal{F}$. Then (i) if $\gamma \ge 1$, $\mathcal{F}$ is un-reconstructible; (ii) if $\gamma \le 0.252$, the* Crust *of S doesn't contain any edge between nonadjacent sample vertices on the original curve $\mathcal{F}$.*

Lemma 1 implies that given the sampled points S, there may not be a unique graph on S that connects every pair of samples adjacent along that graph when $\gamma \geq 1$. In addition, when $\gamma \leq 0.252$, all the piece-wise-linearly-connected edges in the *Crust* "belong" to the original curve $\mathcal{F}$.

LEMMA 2. *(LBVA) For a γ-sampled curve by S in the plane with $\gamma < 1$, the Voronoi Angle (i.e., $\angle s_1 v s_2$ in Fig. 2) formed at a curve Voronoi vertex v between two adjacent samples along $\mathcal{F}$ is at least $\pi - 2\arcsin(\frac{\gamma}{2})$.*

The Voronoi Angle physically represents the discrepancy between the real curve and the recovered curve. And the higher of this angle, the smaller of such discrepancy. Ideally, the case of $\angle s_1 v s_2 = \pi$ means that the recovered curve precisely matches the original one. We sketch a proof here. When $\angle s_1 v s_2 = \pi$, in the original curve $\mathcal{F}$ should have a straight-line connecting $s_1 s_2$. Otherwise, it will have small local feature size for some points between s_1 and s_2. Then the sampling condition will imply that we should have additional sampling points between s_1 and s_2, which contradicts the assumption that s_1 and s_2 are consecutive samples.

Intuitively, the more of sample points, the better of curve reconstruction quality. However, the larger size of S generally leads to the increase of sampling cost. Note that as the presence of the strong dependence among the entries in S, the marginal gains of LBVA might be significantly small and negligible as the increase of S. Therefore, S must be chosen carefully: it is desirable to have LBVA as high as possible to guarantee the quality of estimated curve, and minimize the cost and privacy leakage caused by collecting more points.

2.3 Alternative Objectives and Challenges

Recall that the task of map generation is to construct one unobserved map from collected GPS locations shared by a variety of users. In a sense, the map could be reviewed as one curve in the 2D plane. Therefore, our problem can immediately fall into the category of curve reconstruction. Here the discrete sampled points correspond to the shared GPS locations. And *Crust* could be used as the map estimate procedure.

According to Lemma 2, the quality of recovered curve could be indirectly measured by the Lower Bound of Voronoi Angle. Therefore, problem **P** could be reformulated as

$$\textbf{(P)} \quad \cup_{i=1}^{m} P_i^* = \underset{\forall i, P_i \subseteq S_i}{\arg\max} \quad \Gamma\left(\cup_{i=1}^{m} P_i\right)$$
$$\text{subject to} \quad PE_i(P_i) \leq b_i,\ 1 \leq i \leq m,$$

where $\Gamma(.)$ returns the Lower Bound of Voronoi Angle(*i.e.*, $\pi - 2\arcsin(\frac{\gamma}{2})$). Unless otherwise specified, this problem will be referred to as MaxLBVA for simplicity purpose in our subsequent discussion.

There are however three major challenges in applying *Crust* to our problem context. First, under the curve reconstruction framework the set of sampled points is exactly from the original smooth curve. However, in the physical environment each road has certain width which determines the distribution of the reported GPS locations will be arbitrary within that road, instead of along one smooth curve that we expect. In particular, for a two-way road with four lines, the *Crust* might infer the existence of one road between the points from different lines even if they are indeed physically from the same road. This makes it difficult to construct one high-quality map via directly using *Crust* on the raw collected data.

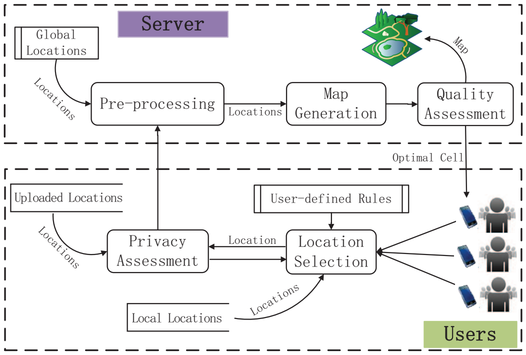


Fig. 3: The architecture of our scheme PMG

A second challenge is that GPS data is not error free. The users can simply suppress the data if the error exceeds a predefined threshold. However, doing this might not completely remove all potential errors. This is because some other factors such as local obstructions, weather and users' movement pattern might also degrade the GPS performance.

At last, MaxLBVA is a combinatorial problem with linear constraints, which has been shown in [22] to be NP-complete. A simple greedy algorithm is often used instead. It has a $O(1)$ approximation ratio with a submodular objective. However, compared with traditional combinatorial problem, on one hand, for map reconstruction we do not have direct access to the set of all sampled points collected at users' sides; on the other hand, solving this problem could only be finished in the decentralized framework that involves in extra coordination between the users and the server. We show in subsequent section how these challenges can be addressed in our scheme such that we can implement this simple yet effective heuristic in our specific context.

3 Proposed Solution

3.1 System Architecture

Figure 3 shows the overall architecture of our solution. At the network level, the system consists of a number of users with elegant privacy-preserving schemes, who would like to contribute their locations and a map generation server

Users: The users serve as the GPS location provider. To provide certain diversity of uploaded data, one finite local buffer is used to record the user's trace. One data report engine, called Location Selection, would be activated by the location query from the remote server.

Once receiving such *request* packet, the users will lookup their corresponding local buffer and reply the server with the locations that match the request condition. More information about the *request* packet will be discussed in Section 3.3. To avoid potential trace leakage, all the reported locations must go through one privacy-assessment module. As a result, only the "safe" data will be allowed to stream into the server.

Server: The essential function of the server is to provide high-quality map generation service based on the collected unorganized GPS locations from various users. To guarantee the estimated map quality, all chosen GPS locations will firstly enter into one data pre-processing block to remove all unjustified data. Then, only the valid data will go into the map generation module, which has implemented the aforementioned *Crust* algorithm. The following module, called Quality Assessment, is then executed to examine the quality of current generated map (*i.e.*, the output of *Crust*). When the predefined map quality metric is not met, the block is further scheduled to estimate the optimal cell that will provide maximal gains in estimating the original map; server will broadcast this cell via *request* packet to *actively* pull the useful information. One practical optimal location selection will be introduced and analyzed in Section 3.3.

3.2 Privacy-preserving Scheme

3.2.1 Threat model and trace privacy-preserving

Threat Model: Since our system consists of multiple shared clients and a server, any adversary is able to overhear the packets transmitted between them or maliciously access the server to obtain the shared data. Thus, the threat model we consider is the attacks on the server and its corresponding communications with the shared clients. For simplicity purpose, we assume the data stored on each client are safe and unobtainable. And we further assume that the adversaries' attack objective is to infer users' personal sensitive information (*e.g.*, home address, workplace, behavior pattern or health condition) or trace through analyzing or mining the obtained data. Therefore, we mainly focus on the risks of privacy leakage based on the shared data.

Trace Privacy-preserving: Although many participatory sensing applications have been implemented to generate/update the unobserved map accurately based on shared users' traces [2] [24], such trace based applications would inevitably lead to users' privacy leakage. Therefore, trace protection should avoid to disclose the shared users' trajectory as well as to speculate other personal sensitive information. In our map generation application, one main goal of trace privacy-preserving is to protect individual users' trajectory within a specific time interval, formally described as follows.

Definition 5. *(IUT: Individual User's Trajectory/Trace) Consider a time interval T, one trajectory of an individual user in the $2D$ space is defined as a sequence of tuples $IUT = \{\langle p_1^x, p_1^y, t_1\rangle, \langle p_2^x, p_2^y, t_2\rangle, \dots, \langle p_n^x, p_n^y, t_n\rangle\}$ where p_i^x and $p_i^y (i = 1, 2, \dots, n)$ mean the GPS coordinates of location p_i and $1 \leq t_1 < t_2 < \cdots < t_n \leq T$ are ordered discrete time instances within a time interval of T.*

3.2.2 Privacy-preserving policy

Within one time interval, it is well understood that the leakage of user's traces greatly depends on the amount of shared locations, thus one natural way is to define b_i as the maximal number of uploaded locations set by user i and $PE_i(.)$ as the actual number of uploaded locations from user i. Using this simple rule, the user will significantly reduce the locations reported to the server, thus the degree of trace exposure is also decreased. However, it should be noted that since location data with time dimension are highly correlative, some attack strategies could easily infer user's traces only with a few locations. Therefore more advanced and elegant privacy-preserving policy is highly expected.

Consider a time interval T. Our concept is to provide a group of unorganized locations which might correspond to various quite different routes. In other words, given the shared locations, no algorithm could uniquely and easily determine the real route that he has been passing. Mathematically, this curve reconstruction problem is unsolvable. Therefore, besides breaking the temporal relationship among reported locations, we should also focus on how to choose appropriate locations to avoid leaking the spatial relationship among these locations. Specifically, the client will use another finite buffer to store all the points that have been reported within T. Once having another new reported location point, the user will examine the trace leakage degree, which corresponds to the privacy-exposure function $PE_i(.)$ in problem **P**, to decide whether reporting this point to the server. This new point is said to be *safe* or *qualified* if the return value of $PE_i(.)$ is less than b_i. Next, we will discuss how to choose the privacy-exposure function $PE_i(.)$ and the upper cost bound b_i.

From trace protection prospective, the most effective metric is to let $PE_i(.)$ return the deviation between estimated trace of the reported points and the real trace. Clearly, the higher value of this metric, the better of the trace protection would be. Ideally, the user's trace could be completely protected when the value of deviation is nearly infinite. However, it is impractical for a user to record all the GPS locations within T, so the real trace is incomplete which means it is impossible to compute this deviation evenly. Here, we bypass the deviation metric and choose γ-sample to indirectly measure the trace exposure degree. In Lemma 1, the most effective algorithm *Crust* can't uniquely determine the original curve when γ-sample is more than and equal to 1. From the view of privacy protection, we therefore prefer the locations set to be at least 1-sample, *i.e.*, $\gamma \geq 1$. For instance, if a user wants to exactly protect his/her trace within T, he/she must set γ to be at least 1. From LEMMA 1, a mediate trace protection degree could be achieved through setting γ between 0.252 and 1. And a user's trace would be exposed when $\gamma \leq 0.252$. Note that the smaller γ corresponds to more shared GPS points by the users and better estimated map quality on the server, and vice versa. Obviously, γ reflects the tradeoff between the local trace protection degree and remote estimated map quality. Therefore, in the practical implementation, users could adjust γ based on their specific requirements.

3.2.3 Other issues

Note that there still exist trace exposure for the above mentioned schemes. This is because that the goal of privacy assessment is to protect the user's private trace within T, no guarantee about the larger time interval. For instances, if T is set to be one hour, the user's trace within larger than one hour might be easily determined if the user repeats the trace which he/she has walked, even though the real trace in one hour (even less than one hour) is unsolvable. To maximally avoid the users trace leak, it is desirable to set a large T, which is more likely to have no overlapping between different time windows. Therefore, T reflects the degree of privacy protection and the user should carefully set this value according to his/her privacy requirement. Unfortunately, due to the periodical property of the regular home-office or home-school route, the privacy could not be guaranteed no matter how large T we choose. Therefore, in this paper, we assume the points shared by individual user are sampled from the not-often passed roads.

In addition, some locations that are very sensitive or specific could really identify a person, a trajectory or other private information. Considering this, some extra user-defined rules shown as follows could also be integrated into our application (in Fig. 3).

Rule based on sensitive regions. Individual users could restrict uploading the locations in some sensitive regions, such as home area.

Rule based on sensitive time. Individual users could restrict uploading locations collected in some certain periods of time, such as working hours.

After applying such rules, the privacy-preserving scheme in our framework will not only protect the shared user's trace information, but also avoid the leakage of some potential sensitive information.

3.3 Near-optimal Locations Selection

In this subsection we mainly focus on how and where to query users for locations so as to maximize LBVA. Due to the hardness solving **P** directly, we reformulate it as one equivalent maximization problem over a group of cells. We then demonstrate that the new objective exhibits the property of sub-modularity. One simple greedy algorithm within constant ($\approx 63\%$) of the optimum is proposed.

3.3.1 Proposed alternative formulation

Obviously, MaxLBVA remains the combinatorial optimization which is intractable. Specifically, without any prior knowledge about the map that we wish to estimate, the server is oblivious of the potential location candidates, let alone choosing optimal location set to maximize $\Gamma(\cdot)$ without breaching certain privacy constraints. Finding the optimal solution of MaxLBVA is non-trivial, especially when any location in the space could be the candidate.

To bridge the gap between server's difficulty of having little knowledge on where to pick points to improve map quality, and the points that have been collected by individual users, we will partition the region into a group of continuous cells. Specifically, given historical knowledge and collected locations, the server firstly sets up the general region in which the map need to be generated or updated and then divides this region into $n \times m$ continuous squares (*i.e.*, cells). Next, the server estimates marginal gain (*i.e.*, the improvement of the map quality if it asks for points from users) of each cell. It picks the cell with the best marginal gain and asks users to report locations inside this cell. Assume a region is divided into w cells and use a complete set $I = \{1, 2, \cdots, w\}$ to denote them. Instead of seeking for exact locations set, we alternatively look for a subset of I, each cell possibly including infinite location points. Therefore, we reformulate **P** as follow

$$\textbf{(P)} \qquad A^* = \underset{A \subseteq I}{\arg\max} \quad \mathcal{R}(A) = \mathbb{E}[\Gamma(A)] - \Gamma(S_0)$$

where $\mathbb{E}[\cdot]$ denotes expectation operation, computed over all locations uniformly distributed within the cells and $\mathcal{R}(\cdot)$ represents the margin gain of the cells, that is the improvement of map quality.

The function $\Gamma(\cdot)$ we defined is over the location set, rather than cells. Here we approximate $\Gamma(A)$ as the expectation of LBVA when we query points from a set of cells A. Since we have no idea about the underlying road distribution and users's movement pattern, it is reasonable to assume that the each location within a cell will be uniformly reported. If $A = \{a_1, a_2, \cdots, a_{|A|}\}$, then $\mathbb{E}[\Gamma(A)]$ is computed sequentially

$$\mathbb{E}[\Gamma(A)] = \sum_{i=1,a_i}^{|A|} \int_{\forall p \in a_i} \frac{1}{r^2} \Gamma(S_{i-1} \cup p), \qquad (1)$$

where r is the side length of a cell, p is a location in the cell and S_i is the collected location set by the server after choosing the i-th cell. Here S_0 means the initial sporadic collected locations by the server and is used as the base of optimal points/cell query and points update rather than groundtruth. When no cell is chosen (*i.e.*, $A = \emptyset$), the expectation is only determined by S_0. Thus we have $\mathbb{E}[\Gamma(\emptyset)] = \Gamma(S_0)$ and $\mathcal{R}(\emptyset)$=0.

Regarding the cost associated with each cell, it could be interpreted as the time interval spent to grab locations data. Once a cell was chosen, we wish more users in such cell could quickly reply the server. However, it is impossible to estimate such time duration due to the separated procedure of cell selection and cost estimation. Given a cell, intuitively, the more of qualified users, the quicker of collecting locations. For each cell, therefore, if there is at least one qualified user, the corresponding cell cost is set to be one over the total number of qualified users, and infinity for otherwise.

3.3.2 Properties of the objective

There are several important and intuitive properties of $\mathcal{R}(A)$. Firstly, as mentioned, we have $\mathcal{R}(\emptyset) = 0$. Secondly, $\mathcal{R}(A)$ is nondecreasing. That is $\mathcal{R}(A) \leq \mathcal{R}(B)$ for all cell subsets $A \subseteq B \subseteq I$. Clearly, adding more cells means that more points will be chosen, thus incurring the improvement of the LBVA and estimated map quality. Therefore, choosing more cells will further incur the increase of $\mathcal{R}(A)$. Last but most importantly, it exhibits diminishing marginal returns. To be specific, adding a cell to a small subset A, the reward that we can obtain would be at least as much as if adding it to a larger one $B \supseteq A$, which is implied formally by the following theorems.

Theorem 1. *Consider a smooth curve $\mathcal{F}$. Let V to be the universal points set. For all $S_1 \subseteq S_2 \subseteq V$ and all points $p \in V \setminus S_2$, it holds that*

$$\gamma(S_1 \cup p) - \gamma(S_1) \geq \gamma(S_2 \cup p) - \gamma(S_2),$$

where the function $\gamma(S_1)$ returns the sample condition of S_1 on $\mathcal{F}$. A set function with this property is called sub-modular.

Proof: For any $p \in V \setminus S_2$, denote its two nearest neighbors in S_2 as p_1 and p_2, respectively. According to Definition 3, adding a point will definitely affect some points' Euclidean distance to their corresponding nearest neighbors in the sample set. And these points will be referred to as affected points below. Our proof is similar to the analysis of [31]. We have the following three cases:

Case 1: $p_1, p_2 \in S_1$: Since $p_1, p_2 \in S_1$, the affected points are exactly same for S_1 and S_2. Given a smooth curve, the local feature size of any point is invariable. From Definition 3, we have that the new added point p will lead to exactly same gains, *i.e.*, $\gamma(S_1 \cup p) - \gamma(S_1) = \gamma(S_2 \cup p) - \gamma(S_2)$.

Case 2: $p_1, p_2 \in S_2 \setminus S_1$: For the set S_2, the affected points are a subset between p_1 and p_2. Since $p_1, p_2 \notin S_1$, the affected points at least contain all the points between p_1 and p_2. Therefore, we have $\gamma(S_1 \cup p) - \gamma(S_1) > \gamma(S_2 \cup p) - \gamma(S_2)$.

Case 3: $p_1 \in S_1, p_2 \in S_2$ (or $p_2 \in S_1, p_1 \in S_2$) When $p_1 \in S_1$, this means that $p_2 \in S_2$. Thus, the affected points of S_2 are a subset between p_1 and p_2. Since $p_2 \notin S_1$, the number of S_1's affected points must be more than that of B. Thus, we have $\gamma(S_1 \cup p) - \gamma(S_1) > \gamma(S_2 \cup p) - \gamma(S_2)$.

We thus conclude that $\gamma(S_1 \cup p) - \gamma(S_1) \geq \gamma(S_2 \cup p) - \gamma(S_2)$ and the function $\gamma(\cdot)$ is submodular. □

Theorem 2. *$\mathcal{R}(\cdot)$ is submodular set function.*

Proof: Since $\Gamma(S) = \pi - 2\arcsin\frac{\gamma(S)}{2}$ and $\gamma(\cdot)$ is submodular, we have $\Gamma(\cdot)$ is submodular too. From Section 3.3.1, we know that $\mathcal{R}(A)$ is the integration of $\Gamma(\cdot)$ over all possible points in a cell. Therefore, $\mathcal{R}(\cdot)$ is submodular as well. □

3.3.3 Proposed greedy algorithm

In general, maximizing submodular functions is NP-hard [22]. We instead use a heuristic greedy algorithm to obtain a sub-optimal solution.

The simple one is the unit cost case, where each cell has equal unit cost (*i.e.*, for any cell i, $c(i) = 1$). The greedy algorithm will reduce to select b cells from I with the highest score. It operates as follows: starting from $A = \emptyset$, iteratively add a single cell with the highest score, conditioned on the cells chosen in previous steps until the map quality reaches a certain threshold. More formally, at each step, the greedy algorithm adds the element cell i such that

$$i^* = \arg\max_{i \in I \setminus A} \mathcal{R}(A \cup i) - \mathcal{R}(A). \tag{2}$$

At each step, the optimal cell could be immediately determined by the Eq. (1) and (2). Next the server broadcasts one *request* packet containing the physical information (*e.g.*, GPS coordinate for the cell's four corner points) of the chosen cell. Any user hearing such *request* packet will examine the locations falling in the chosen cell. If the matched location set is nonempty, the privacy assessment scheme is further applied on them to remove all non-safe data that might lead to trace leakage. In response to the *request* packet, eventually, the user will send the final chosen safe locations back to the server.

We end this part by discussing the theoretical bound of our proposed simple greedy algorithm. Firstly, we give one proved conclusion in [22], denoted as the following lemma.

LEMMA 3. *(Nemhauser et al., 1978) Let F be a monotone submodular set function over a finite ground set V with $F(\phi) = 0$. Let A_G be the set of the first k elements chosen by the greedy algorithm, and let $OPT = \max_{A \subset V, |A| = k} F(A)$, Then*

$$F(A_G) \geq \left(1 - \left(\frac{k-1}{k}\right)^k\right) OPT \geq (1 - 1/e) OPT.$$

This lemma shows that if a set function meets certain conditions, the simple heuristic greedy algorithm could achieve a constant-factor ratio to the optimum.

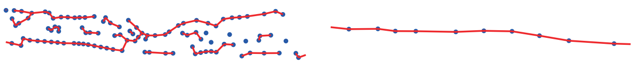
Theorem 3. *Let $\hat{A}$ be the chosen cells by the greedy algorithm and $A^* = \max_{A \subset I} \mathcal{R}(A)$. Then*

$$\mathcal{R}(\hat{A}) \geq (1 - e^{-1}) \mathcal{R}(A^*).$$

Proof: Since $\mathcal{R}(\cdot)$ is a nondecreasing submodular set function with $\mathcal{R}(\emptyset) = 0$, based on LEMMA 3, we have $\mathcal{R}(\hat{A}) \geq (1 - e^{-1}) \mathcal{R}(A^*)$. □

3.4 Impact of GPS Sample Error and Road Width

In our scheme, the location data is mainly from GPS sampling based on smartphones. Therefore, two critical issues must be addressed to make our protocol practical: 1) GPS sample error (thus, samples are not necessarily from the real curve $\mathcal{F}$), and 2) road width (thus, over-sampled points will result in extra small segments). The curve reconstruction problem assumes a smooth curve



(a) Recovery without clustering (b) Recovery with clustering

Fig. 4: Impact of road width.

with zero thickness and the unorganized points precisely from the underlying curve. While in our situation, even if the map could be viewed as a smooth curve, the thickness of each edge could not be zero. The map generation algorithm (*i.e.*, *Crust*) might add extensive unnecessary roads/edges within the same road, especially when the road width is very large (*e.g.*, high way). Figure 4 illustrates an example.

For dealing with GPS error, we first remove the data when such accuracy (*e.g.*, getAccuracy() in Android returns the standard deviation of the GPS measurement in the current location) is more than a threshold η. It is reasonable to set η to be double road width, about $40m$. Even so, the uploaded data is still noisy. We then apply a simple clustering algorithm to the filtered data. Specifically, the collected points will be divided into several clusters based on the locations' geographical proximity. And we use the cluster center to represent a sample from the underlying map. We run the *Crust* algorithm using the cluster centers rather than all collected raw points.

Consider one location in the 2D plane with x_0 as the GPS ground truth. Let $x_1, x_2, \ldots, x_n$ be the measured value by n different users with x_i in a small cluster, which could be seen as the realization of a random variable X with mean x_0. Considering the Gaussian noise, X could be modeled as $X = x_0 + N(0, \sigma^2)$. We next theoretically show that with the increase of the number of reported users, the empirical mean will be close to the real value with higher probability (close to 1).

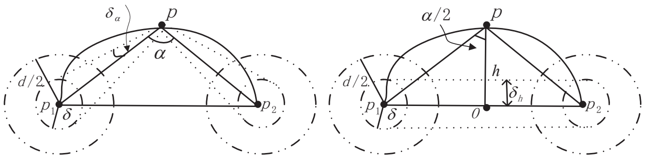
Theorem 4. *Given one location with n real measured noisy value x_i. If $x_i \in [-\frac{d}{2}+x_0, \frac{d}{2}+x_0]$, then we have*

$$Pr(|\frac{1}{n}\sum_{i=1}^{n} x_i - x_0| \geq \delta) \quad \leq \quad 2exp(-2n^{\frac{1}{3}}), \qquad (3)$$

which is valid for positive value of $\delta = \frac{d}{n^{\frac{1}{3}}}$.

Proof: This theorem could be achieved directly based on Hoeffding's inequality: Suppose $X_1, X_2, \ldots, X_n$ are independent real-valued random variables, such that for each i, X_i takes values from the interval $[a_i, b_i]$, Let $\overline{X} = \frac{1}{n}(X_1 + X_2 + + X_n)$. Then, for all $\delta > 0$, $Pr(|\overline{X} - E(\overline{X})| \geq \delta) \leq 2exp(-\frac{2n^2\delta^2}{\sum_{i=1}^{n}(b_i-a_i)^2})$. Here, with $x_i \in [-\frac{d}{2}+x_0, \frac{d}{2}+x_0]$ and $\delta = \frac{d}{n^{\frac{1}{3}}}$, we could naturally achieve Eq.(3). □

Using the collected noisy GPS data, we examine the performance of the Voronoi Angle (*i.e.*, α in Fig. 5(a)) and the maximal Euclidean distance between the real curve and the estimated (*i.e.* h in Fig. 5(b)). Consider two consecutive sample points p_1 and p_2 on a smooth curve $\mathcal{F}$, as shown by Fig. 5. Due to the noise in the physical setting, their corresponding real measured GPS values are actually uniformly distributed within the two bigger dashed circles with radius $\frac{d}{2}$. From lemma 2, we have $\alpha = \pi - 2\arcsin\frac{\gamma}{2}$ under the noise-free assumption.



(a) Voronoi Angle (b) Max distance gap between the real curve and the estimated

Fig. 5: The effect of GPS data error.

From Theorem 4, we know that the sampled GPS data of p_1 and p_2 will concentrate in the two smaller disks with radius δ. Clearly, we can see that their corresponding α and h will fall in the range of $[\alpha - \delta_\alpha, \alpha + \delta_\alpha]$ and $[h - \delta_h, h + \delta_h]$, respectively. Based on the basic geometric knowledge, $\delta_\alpha = \arcsin\frac{\delta\gamma}{\tan(\arcsin\frac{\gamma}{2})Ds(p_1,p_2)}$ and $\delta_h = \delta$.

Clearly, these two metrics quantifying the quality of recovered map will fluctuate within a very small range, determined by δ. Similarly, as the number of samples increases, they will approximate their corresponding ground truth with higher probability (close to 1). This means that our proposed map generation scheme is robust against the inherent noises of GPS data by clustering (sort of resampling by server).

4 Performance evaluation

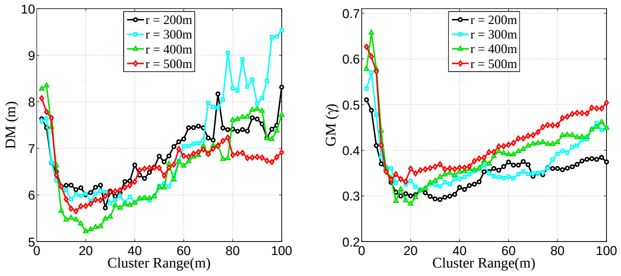
In this section, we present a series of experiments performed on two group city-scale GPS trace data. We focus on the impact of different parameters on the estimated map quality and the overall effectiveness of PMG. We will use greedy algorithm mentioned in section 3.3 to choose optimal cells. The map generator we use is *Crust*.

We will use two datasets. The first one, also referred to as the *Shanghai Data*, is a group of GPS data published on the CrowdAtlas website with 24 traces containing 954000 locations in total [23]. The area of this dataset is about $149.09km^2$ and the total length of traces is $111390m$. The second, referred to as the *Wuxi Data*, was collected in Wuxi New district, with 323120 locations. And its area and traces are $36.45km^2$ and $29284m$, respectively.

Due to the lack of large scale participant sensing filed, we reshuffle the two datasets and randomly assign these locations into m different files to emulate the number of users. This value (*i.e.*, m) is set be 10 and 50 for the Wuxi data and Shanghai data, respectively. In addition, each user defines his/her privacy protection level to be no trace leakage within a day (*i.e.*, $T = 24h$ and $\gamma^{'}$=1).

Denote the recovered segments set as $\hat{E} = \{e_i, 1 \leq i \leq |\hat{E}|\}$, each segment with n^i points. We next will use two metrics to verify the effectiveness of PMG: one is Deviation Metric(DM) denoting how far is the estimated map from the ground truth, and the other is Gamma Metric(GM), an indirect criterion measuring the estimated map quality. They are given by

$$DM = (\sum_{i=1}^{|\hat{E}|} DM_i)/|\hat{E}|, \quad GM = (\sum_{i=1}^{|\hat{E}|} GM_i)/|\hat{E}|$$



(a) DM against cluster range (b) GM against cluster range

Fig. 6: The impact of cluster range.

which DM_i/GM_i is also referred to as segment DM/GM, defined as $DM_i = (\sum_{j=1}^{n^i} h_j^i)/n^i$ and $GM_i = (\sum_{j=1}^{n^i} \gamma_j^i)/n^i$. Here, h_j^i is the j-th point's physical deviation from the true value on e_i and γ_j^i denotes this point's sample condition on segment e_i.

4.1 Impact of different parameters

In this subsection, we observe the impact of different parameters (*i.e.*, cluster range, cell size and the degree of privacy protection) on the final estimated map quality. We conducted our experiments on the two datasets, the effect of which share similar trend. Thus, we only report the results on *Wuxi data*.

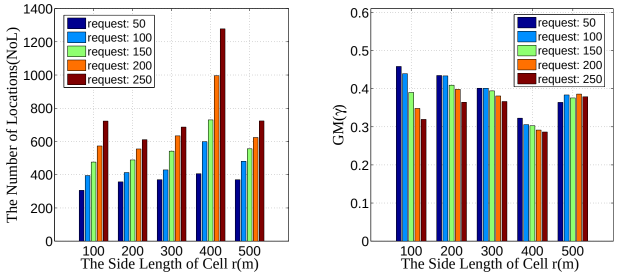
4.1.1 Cluster range

Figure 6 shows the performance of DM and GM by adjusting the cluster range from $0m$ to $100m$, with increments of $2m$. We run this experiment for 4 times using four different side lengths of cell (*i.e.*, $r = 200, 300, 400, 500$, all in units of m).

Regardless of the cell size r, we can clearly see that both DM and GM behave a sharply downward trend at the beginning, then decrease slowly between $15m$ and $25m$ and increase gradually when the cluster range is more than $30m$. Moreover, the quality of the generated map could achieve the empirical optimum/minimum when the cluster range is around $20m$, which is consistent with the real road width (about $20m$). Note that the bigger of the cluster range, the sparser of the collected points. Thus, as the cluster range grows, the real input of our map estimator (*i.e.*, *Crust*) will fail to reflect the road features, such as corner. This is the reason why the performance of DM and GM degrades gradually when the cluster range is more than $30m$.

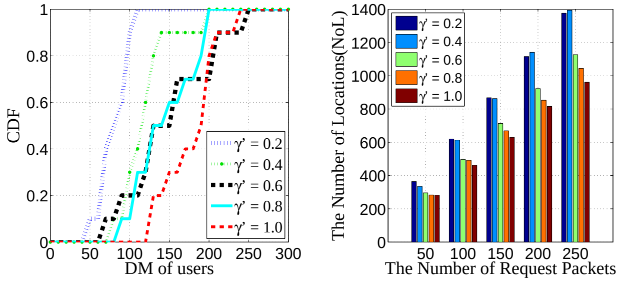
4.1.2 Cell size

We next examine the effect of cell size on the generated map quality. Due to the performance similarity between DM and GM, we only offer the performance of GM under different cell size. Since our location selection algorithm is cell-based, we also investigate the impact of different cell size on the Number of Locations (NoL) (*i.e.*, the number of all real collected locations when the greedy algorithm finishes). We did this experiment under different number of *request* packets from the server. The results are shown in Fig. 7.



(a) Number of Locations (NoL) (b) Recovered map quality GM

Fig. 7: The impact of cell size



(a) CDF of users' deviation (b) Number of locations in server

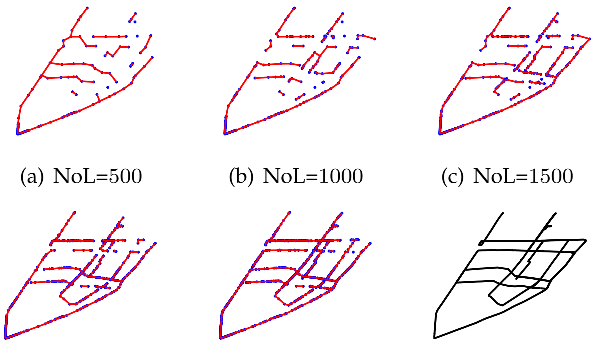
Fig. 8: Influences of different degree of privacy protection

From Fig. 7(a), we can see that as the increase of r, NoL increases at first, achieves a peak when $r = 400$, then begins to decrease. GM behaves the opposite trend. The result is reasonable. When cell size is small, each cell might contain a few matched locations, so after hearing the *request* packet, less users will response. When r is increasing, more qualified locations might be contained in each cell, leading to the increases of NoL. However, when $r = 500$, the cell size will be so large that the chosen cell might contain many roadless areas, which results in the worse performance relative to $r = 400$.

Once the cell size is fixed, the NoL(GM) behaves monotonically increasing(decreasing) with the increase of the number of *request* packets. This is because more *request* packets mean more collected locations, which improves the final generated map quality (*i.e.*, the decrease of GM). However, there is a small exception for GM when $r = 500$ (see, Fig. 7(b)). Again this is due to that r is too large, containing many areas without roads which might lead to the less number of qualified locations.

4.1.3 The degree of privacy protection

In Lemma 1 and Section 3.2, we have pointed out that $\gamma^{'}$ in client reflects the degree of privacy protection and the user could set this value according to his/her privacy requirement. If $\gamma^{'}$ is set to be 1.0 , it means that γ computed by the reported locations of individual users within T must be bigger than $\gamma^{'}$, *i.e.*, $\gamma > 1.0$. Theoretically, for individual users, a higher value of $\gamma^{'}$ in client is better to preserve his/her privacy. This is also clearly shown by Fig. 8(a). In Fig. 8(a), we can find that with the increase of $\gamma^{'}$, the deviation between the estimated traces of individual users and the ground truth is increasing. When $\gamma^{'} = 0.4$, DM of most users is more than $100m$ and when $\gamma^{'} = 1.0$, even all users have the


(a) NoL=500 (b) NoL=1000 (c) NoL=1500

(d) NoL=2000 (e) NoL=2500 (f) Groundtruth

Fig. 9: *Wuxi data*: generated maps at different NoL

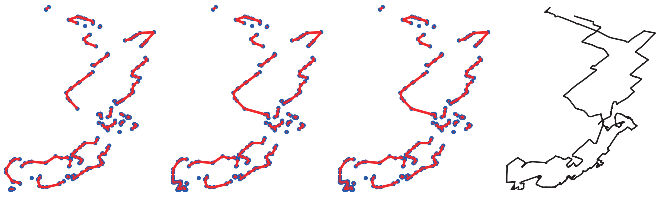

(a) NoL=500 (b) NoL=1000 (c) NoL=5000 (d) Groundtruth

Fig. 10: *Shanghai data*: generated maps at different NoL

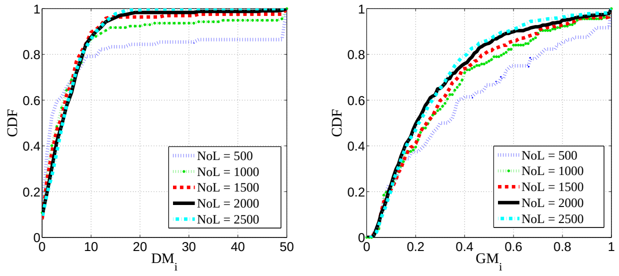
DM of more than $100m$.

For the server, however, a higher value of $\gamma^{'}$ in client is not better for the efficiency of map generation. From Fig. 8(b), we could see that when the number of *request* packets received by each user is the same, the bigger $\gamma^{'}$ is, the less the number of locations uploaded to the server is. This means more time is needed to generate or update a map if a bigger $\gamma^{'}$ is set. Another meaningful discovery is that when $\gamma^{'} = 0.2$ and 0.4, the number of locations received in server is nearly the same, while when $\gamma^{'} = 0.6$, 0.8 and 1.0, the number is significantly fewer. This is because individual users only send a few locations to the server in a period of time, *e.g.*, one day, which makes the traces generated by these locations in client quite different from ground truth, *i.e.*, γ computed in client is usually bigger than the set threshold $\gamma^{'}$. Therefore, when the set threshold is low, *e.g.*, 0.2 or 0.4, the user usually could response most *request* packets if he/she has the satisfied locations. With the increase of the set threshold, however, more points are not satisfied this stricter demand of privacy protection. Hence, these points would not be sent to the server so that the number of locations received in server decreases.

By analyzing and comparing these two figures in Fig. 8, it is obvious that in a real environment, an appropriate threshold $\gamma^{'}$ is essential for both individual users and the server. Based on the result shown in Fig. 8, it is relatively appropriate to set $\gamma^{'}$ to be from 0.6 to 1.0.

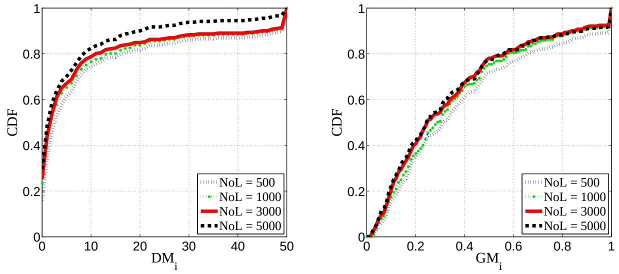
4.2 The quality of generated map

In this section, we will investigate the generated map quality in various dimensions. Unless otherwise stated, we will set the cluster range and size length of a cell r to be $20m$ and $400m$ in the next experiments.



(a) CDF of segment DM (b) CDF of segment GM

Fig. 11: CDF observation with *Wuxi data*



(a) CDF of segment DM (b) CDF of segment GM

Fig. 12: CDF observation with *Shanghai data*

TABLE 1: Map Generation Results with Two Data Sets

DataSet	NoL	NoLC	DM	$\lvert\hat{E}\rvert$	Length	Density
Wuxi	500	221	128.40m	96	3776.9m	17.09
	1000	230	38.44m	157	4941.1m	21.48
	1500	273	25.31m	256	8153.9m	29.87
	2000	432	6.04m	416	13430m	31.09
	2500	594	5.53m	578	20186m	33.98
Shanghai	500	273	8.93m	246	7041.3m	25.79
	1000	304	8.01m	277	6424.2m	21.13
	2000	310	7.79m	284	6335.6m	20.44
	3000	320	7.43m	292	7100.6m	22.19
	4000	326	7.34m	299	7893.0m	24.21
	5000	332	7.38m	307	7945.6m	23.93

4.2.1 Visual comparison

We first visually observe the generated map quality under different sampling points both in Wuxi and Shanghai(Fig. 9 and 10). Here the red lines mean the recovered segments, the blue points represent the clustered sampling locations. And we also provide the ground truth in black (Fig. 9(f) and 10(d)). As expected, the more of sampling locations, the better of recovered map quality. In Wuxi experiment, Figure 9 shows that when NoL $= 2500$, the recovered map could almost capture the general trend of the original map. For the Shanghai data, the performance improvement is very small if changing the number of locations from 500 to 1000. Moreover, such improvement almost disappears when NoL > 1000.

4.2.2 Quantitative evaluation

To be more precise in comparison, we further observe the CDF of segment DM and GM under different number of sampling locations shown in Fig. 11 and 12. The results suggest that the estimated map based on Wuxi data performs better than Shanghai's, *e.g.*, when NoL= 1500, about 90% of the recovered segments are at most $10m$ apart from the ground truth, while for Shanghai, there are only 80% such segments even if NoL= 3000.

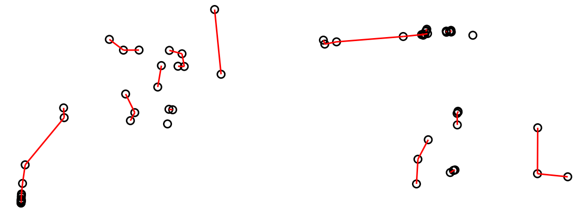
More statistical information describing the recovered map quality is presented in Table 1. Here, NoLC means the number of locations after clustering and Density (with the unit of cluster/m) represents the average distance of consecutive clustered points. The recovered map quality improves as the increase of sampling locations. Such improvement could also be verified by the increase of Density and the decrease of DM. Compared with Wuxi data, the performance gains are not obvious for the Shanghai data. Only 12 new clustered points are added even if NoL adjusts from 3000 to 5000. This is because the number of users (*i.e.*, 50) might be a little big for the cell with side length $400m$. When one request is sent, if too many users response, the sampling locations are too dense, which results in the less NoLC and the slow growth of the quality. Therefore, it is highly necessary to select appropriate parameters based on real situation, *e.g.*, the number of users, the cell side length and so on.

4.3 Evaluation of Privacy Protection

We examine the performance of privacy protection by observing the individual recovered trace. We set the privacy protection level (*i.e.* $\gamma^{'}$) to be one which has been discussed in Section 4.1.3. For simplicity purpose, each user exploits exactly same $\gamma^{'}$. We randomly choose a user's reported locations from *Wuxi data* and *Shanghai data*, then use *Crust* to estimate their corresponding trace within a day. Figure 13 illustrates the recovered individual trace. Clearly, compared with the ground truth in Fig. 9(f) and Fig. 10(d), these two graphs contain so many separated segments and points that we cannot know the users' complete or real trace.

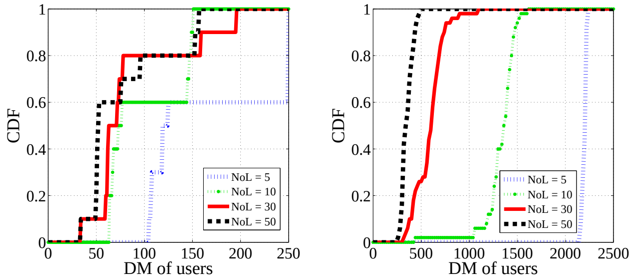
We also observe the CDF of individual users' DM to have a more precise evaluation of privacy protection. Figure 14 shows that when individual users send 5, 10, 30 and 50 locations to the server in one day, the deviation between generated trace of individual users and ground truth is always more than $30m$ with Wuxi users' data and $250m$ with Shanghai users' data. With the increase of reported locations, the deviation is decreasing. Compared with Fig. 14(b), we find that Fig. 14(a) has a much less deviation. This is because Shanghai's data were collected in more complicated roads, so when individual users only report a few locations, it is much more difficult to speculate accurate users' traces. This discovery inspires us that our privacy-preserving scheme might be improved by setting $\gamma^{'}$ based on different road conditions in the future work.

In summary, Figure 13 and 14 demonstrate that even if the server can effectively and accurately recover the unobserved map of individual users, it is impossible for adversaries to infer each user's private accurate trace because the large DM indicates the failure of the $Crust$.



(a) Wuxi user data (b) Shanghai user data

Fig. 13: Recovered trace by one user in one day ($\gamma^{'} = 1$)



(a) CDF of individual users' deviation with *Wuxi data*

(b) CDF of individual users' deviation with *Shanghai data*

Fig. 14: CDF with *Wuxi* and *Shanghai* users' data

5 Related Work

Map Generation and Curve Reconstruction. Nowadays, many mapping projects with crowdsourcing activities have been successfully implemented, *e.g.*, OpenStreetMap [24] and Google Map Maker. Wang *et al.* develop an application CrowdAtlas [2] to automate map update based on user traces. When there are sufficient traces, map inference algorithms can automatically update the map. Shen *et al.* present Walkie-Markie [3] to generate an indoor map based on user trajectories and use WiFi-Marks based on the RSS trend to locate. Although these approaches could generate maps efficiently, none of them consider protecting user privacy. Therefore, we propose to generate a map with unorganized points uploaded by users instead of the whole traces.

Our idea is coincident with curve reconstruction methods in computational geometry. α-shape is one representative work and uniquely determines a polytope by a finite point set and a parameter α. However, the parameter α must be chosen experimentally and is constant during the recovery, while in map generation there is no ideal value of α due to the unconstant sampling density. In γ-neighborhood graph [25], the sampling density should be the same in each part of the curve, so it isn't suit for the map generation either. β-Skeleton [26] is similar to γ-neighborhood graph except that the radius of the $forbidden\ region$ of two points in β-Skeleton is the same while in γ-neighborhood graph is different. Furthermore, like α-shape, β-Skeleton also needs to choose an appropriate threshold β to ensure the results of curve reconstruction. While in our work, we use $Crust$ [16], which has been introduced in detail in section 2.2, to reconstruct the map as well as to evaluate the privacy leakage because of its simplicity, theoretical guarantees and good estimated quality.

Privacy Protection: Privacy protection is an important issue in participatory sensing and numerous protection strategies have been proposed (*e.g.* see a well survey from [40]). According to the protection method, they could be roughly divided into data suppression, data perturbation, anonymization/ pseudonymity, and data aggregation. Below we will review them in detail.

Data suppression. The idea of this strategy is to control data collection or publication. For example, users could selectively collect or report certain locations/traces based on some predefined rules [27]. Literature [44] iteratively suppresses selected locations from the original traces until a privacy constraint is satisfied. Gruteser *et al.* [45] differ sensitive area and insensitive area in a map. Once users enter into a sensitive area, location updates are delayed or not released from that area and vice versa.

Data perturbation. As the name suggests, the collected data will be distorted by adding an artificial noise before publishing out to the sever [32]–[34] . Ganti *et al.* [33] propose to generate a noise model with characteristics similar to a realistic dataset by using an approximate model of the phenomenon monitored by the application. Pham *et al.* [32] develop a correlated noise model that can be utilized to perturb location-tagged data to protect both the data and location privacy(especially multidimensional correlated time-series data) while allowing community statistics to be reconstructed accurately. Mir *et al.* [34] propose DP-WHERE to achieve differential privacy by adding controlled noise to the set of empirical probability distributions.

Anonymization. The most widely approach of anonymization is k-anonymity [28], the basic idea of which is to remove some features such that each record is not distinguishable among other $k-1$ items. Terrovitis *et al.* [44] have defined a new version of the k-anonymity guarantee, the k^m-anonymity which relies on generalization, to limit the effects of data dimensionality. Beresford *et al.* [29] [37] design mix zone model which assigns users in mix zones different pseudonyms to hide their paths. Delphine *et al.* [41] propose an anonymity-preserving reputation framework based on blind signatures which use periodic pseudonyms to prevent an adversary from compromising the user reports to extract private information.

Data Aggregation. This method relies on a mutual protection within participants rather than applications or a third party to protect data privacy. The collaborative path hiding mechanism [42] proposes that users exchange previously collected sensor readings when they physically encounter each other. Swapping a subset of the data samples removes the association between the sensor readings and the identity of the users who collected them so that the sensor readings don't reveal the actual paths of each user.

5.1 Comparison with Prior Work

The goal of our work is to generate an accurate and reliable map while avoid leaking the users' privacy. To this end, we now discuss the limitations of prior work about privacy protection under map generation and compare our method with them. We summarize the differences in TABLE 2.

TABLE 2: Comparison with Existing Work

Methods	**TP**[1]	**DV**[1]	**DD**[1]	**MQ**[1]
Suppression [14] [27]	Low	Normal	×	Uncertain
Perturbation [33]	Medium	Normal	√	Poor
Anonymization & Pseudonymity [28] [41]	Medium	Normal	√	High
Aggregation [42]	Medium	Large	×	High
PMG	High	Small	×	High

Data suppression is a simple and effective strategy that is most related to our work. Most methods, however, rely on some unsophisticated rules so that they could only protect parts of locations but not guarantee the safety of the whole trajectory [14] [27]. Moreover, once the background of adversaries and users is unknown or the suppression is excessive or unreasonable, this strategy is no longer applicable. In our work, each user merely needs to contribute little data, *e.g.,* less than 100 locations rather than the whole trace. Meanwhile, in order to increase the utilization of each shared locations, we also design a greedy algorithm to maximize the marginal benefits of each location so that the map could achieve a constant factor of the optimum as soon as possible. Therefore, our method could generate a high-quality map even with suppressing most locations of the traces.

Additionally, other privacy-preserving strategies mentioned above focus on slightly related problems. Data perturbation requires preliminary knowledge about data distribution. It also compromises data authenticity so that it could seriously damage the accuracy and efficiency of map generation, which is also the limitation of pseudonymity. The k-anonymity model based on traces is quite different from that of locations. Accurately, the k-anonymity model based on traces is to ensure that each trace is not distinguishable from other k traces in the anonymous set, which means all sample locations in each trace should be anonymized within this set. As anonymous traces are dynamic, it is a challenging problem that how to determine the anonymous set of the traces. And it is like mix zone that there must be enough users to enhance locations and trace privacy. Our application, however, doesn't require the user density they claim and the predictable user behavior pattern. In data aggregation, the traces could still be leaked if the data is intercepted by the adversaries, or the users and the server conspire to reconstruct certain users' traces. Moreover, transmitting data could result in extra energy consumption which is another vital issue in participatory sensing. However, compared with other strategies, the data volume in our approach during the transmission and used to generate map is greatly reduced.

Compared with the earlier version of this work [46], we clearly point out the formal definition of trace privacy

[1] TP: Trace Protection; DV: Data Volume; DD: Data Distortion; MQ: Map Quality.

and the threat model in the context of map generation. Then we investigate the impact of local privacy protection requirement (*i.e.*, $\gamma^{'}$) on both the users' privacy protection degree and the estimated map quality on the server. This experiment proves that smaller $\gamma^{'}$ could lead to smaller local DM which means weaker trace protection. What's more, the additional result shown in Fig. 14 demonstrates that even if shared locations are unencrypted, the user's private trace during a period T could be effectively protected. In addition, the underlying reason of the performance differences on the two datasets could guide us to explore more sophisticated privacy-preserving scheme in map generation in the future.

6 Conclusion and Future Work

In this paper, we jointly studied the high-quality map generation and the policy of privacy-preserving in the context of participant sensing. We viewed the map as a smooth curve in the 2D plane and leveraged the process of constructing *Crust* to be the map estimator. Based on the γ-sample condition of the *Crust*, our scheme meets the individual user privacy demand and is robust to inherent noises of GPS data. Through extensive numerical experiments over two real city-scale GPS datasets, we showed that the server can generate a high-quality map with error bounded by $10m$ with a noisy sample point about every $7.5m$.

There are several future issues to pursue. First, we need to design more efficient algorithm to choose the locations with maximal gains in estimating the underlying map. Second, the trade-off between the privacy leak and the overall estimated map quality should be quantity. We also need to design schemes that can recover more detailed road conditions such as one-way or two-way road and the traffic load of different road segments.

7 Acknowledgement

This work is supported in part by the NSFC Major Program under grant No. 61190110, NSFC grants No. 61170213, 61170220 and 61373146, NSFC\RGC Joint Research Scheme No. 61361166009, RFDP grant No. 20121018430, National Science Fund for Excellent Young Scientist No. 61422207 and Guangxi Key Lab of Wireless Wideband Communication & Signal Processing 2012 Open Fund. The research of X. Li is partially supported by NSF CNS-1035894, NSF ECCS-1247944, NSF ECCS-1343306, NSF CMMI-1436786, NSFC under Grant No. 61170216, No. 61228202 and No. 61373146.

References

[1] D. Zhang, T. He, Y. Liu, Y. Gu, F. Ye, R. K. Ganti, and H. Lei, "Acc: Generic On-Demand Accelerations for Neighbor Discovery in Mobile Applications," in *Proc. of ACM SenSys*, 2012.

[2] Y. Wang, X. Liu, H. Wei, G. Forman, C. Chen, and Y. Zhu, "CrowdAtlas: Self-Updating Maps for Cloud and Personal Use," in *Proc. of ACM MobiSys*, 2013.

[3] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-Markie: Indoor Pathway Mapping Made Easy," in *Proc. of USENIX NSDI*, 2013.

[4] W. Huang, Y. Xiong, X.-Y. Li, H. Lin, X. Mao, P. Yang, and Y. Liu, "Accurate Indoor Localization Using Acoustic Direction Finding via Smart Phones," in *Proc. of IEEE INFOCOM* 2014.

[5] L. Zhang, K.-B.Liu, Y.-H. Jiang, X.-Y. Li, Y. Liu, and P. Yang, "Montage: Combine Frames with Movement Continuity for Realtime Multi-User Tracking," in *Proc. of IEEE INFOCOM*, 2014.

[6] C. Bo, X.-Y. Li, T. Jung, X. Mao, Y. Tao, and L. Yao, "SmartLoc: Push the Limit of the Inertial Sensor Based Metropolitan Localization Using Smartphone," in *Proc. of ACM MobiCom*, 2013.

[7] C. Bo, X. Jian, X.-Y. Li, X. Mao, Y. Wang, and F. Li, "You're Driving and Texting: Detecting Drivers Using Personal Smart Phones by Leveraging Inertial Sensors," in *Proc. of ACM MobiCom*, 2013.

[8] X.-Y. Li and T. Jung, "Search Me If You Can: Privacy-Preserving Location Query Service," in *Proc. of IEEE INFOCOM*, 2013.

[9] T. Jung, X.-Y. Li, Z. Wan, and M. Wan, "Privacy Preserving Cloud Data Access with Multi-Authorities," in *Proc. of IEEE INFOCOM*, 2013.

[10] Y. Agarwal and M. Hall, "ProtectMyPrivacy: Detecting and Mitigating Privacy Leaks on iOS Devices Using Crowdsourcing," in *Proc. of ACM MobiSys*, 2013.

[11] D. Christin, C. Rokopf, M. Hollick, "uSafe: A privacy-aware and Participative Mobile Application for Citizen Safety in Urban Environments," *Pervasive and Mobile Computing*, vol.-8, no.-5, pp. 695–707, 2013.

[12] E. De Cristofaro and C. Soriente, "Extended Capabilities for a Privacy-Enhanced Participatory Sensing Infrastructure (PEPSI)," *IEEE Transactions on Information Forensics and Security*, vol.-8, no.-12, pp. 2021–2033, 2013.

[13] L. Zhang, X.-Y. Li, Y. Liu, "Message in a sealed bottle: Privacy preserving friending in social networks," in *Proc. of IEEE ICDCS*, 2013.

[14] M. Götz, S. Nath, and J. Gehrke, "MaskIt: Privately Releasing User Context Streams for Personalized Mobile Applications," in *Proc. of ACM SIGMOD*, 2012.

[15] X. Zhang , X. Gui, F. Tian, S. Yu, and J. An, "Privacy Quantification Model Based on the Bayes Conditional Risk in Location-Based Services," *Tsinghua Science and Technology*, vol. 19, no. 5, pp. 452–462, 2014.

[16] N. Amenta, M. Bern, and D. Eppstein, "The Crust and the β-Skeleton: Combinatorial Curve Reconstruction," *Graphical Models and Image Processing*, vol. 60, no. 2, pp. 125–135, 1998.

[17] L. H. De Figueiredo and J. de Miranda Gomes, "Computational Morphology of Curves," *The Visual Computer*, vol. 11, no. 2, pp. 105–112, 1994.

[18] D. Attali, "r-Regular Shape Reconstructionfrom Unorganized Points," *Computational Geometry*, vol. 10, no. 4, pp. 239–247, 1998.

[19] F. Bernardini and C. L. Bajaj, "Sampling and Reconstructing Manifolds Using Alpha-Shapes," in *Proc. of the 9th Canadian Conference on Computational Geometry*, 1997.

[20] D. J. Mavriplis, "An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness," *Journal of Computational Physics*, vol. 117, no. 1, pp. 90–101, 1995.

[21] M. Iri, K. Murota, and T. Ohya, "A Fast Voronoi-Diagram Algorithm with Applications to Geographical Optimization Problems," in *Proc. of the 11th IFIP Conference on System Modelling and Optimization*, 1984.

[22] S. Khuller, A. Moss, and J. S. Naor, "The Budgeted Maximum Coverage Problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[23] "CrowdAtlas Website," http://grid.sjtu.edu.cn/mapupdate/.

[24] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[25] R. C. Veltkamp, "The γ-Neighborhood Graph," *Computational Geometry*, vol. 1, no. 4, pp. 227–246, 1992.

[26] D. G. Kirkpatrick and J. D. Radke, "A Framework for Computational Morphology," *Computational Geometry* (G. Toussaint, Ed.), pp. 217–248, North-Holland, Amsterdam, 1988.

[27] U. Hengartner and P. Steenkiste, "Protecting Access to People Location Information," in *Proc. of Security in Pervasive Computing*, 2003.

[28] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[29] A. R. Beresford and F. Stajano, "Mix Zones: User Privacy in Location-Aware Services," in *Proc. of IEEE Pervasive Computing and Communications Workshops*, 2004.

[30] W. Du and M. J. Atallah, "Secure Multi-Party Computation Problems and Their Applications: A Review and Open Problems," in *Proc. of ACM Workshop on New Security Paradigms*, 2001.

[31] X. Wu, Y. Wu, "Near-Optimal Working Slots Selection for a Sensor via Submodularity," in *the 5th International Conference on Wireless Communications, Networking and Mobile Computing*, 2009.

[32] N. Pham, R. Ganti, M.Y. Uddin, S. Nath, and T. Abdelzaher, "Privacy-preserving Reconstruction of Multidimensional Data Maps in Vehicular Participatory Sensing," in *Proc. of EWSN*, 2010.

[33] Raghu K. Ganti, Nam Pham, Yu-En Tsai, and Tarek F. Abdelzaher, "PoolView: Stream Privacy for Grassroots Participatory Sensing," in *Proc. of SenSys*, 2008.

[34] Darakhshan J. Mir, Sibren Isaacman, Ramn Cceres, Margaret Martonosi, Rebecca N. Wright, "DP-WHERE: Differentially Private Modeling of Human Mobility," in *Proc. of IEEE Big Data*, 2013.

[35] C. Aggarwal, "On k-Anonymity and the Curse of Dimensionality," in *Proc. of Very Large Data Bases*, 2005.

[36] de Montjoye YA, Hidalgo CA, Verleysen M, and Blondel VD, "Unique in the Crowd: The Privacy Bounds of Human Mobility," in *Nature Sci. Rep.*, 2013.

[37] A. R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.

[38] G. Zhong, I. Goldberg and U. Hengartner, "Louis, Lester and Pierre: Three Protocols for Location Privacy," in *Proc. of the 7th Workshop on Privacy Enhancing Technologies*, 2007.

[39] M. Terrovitis, N. Mamoulis and P. Kalnis, "Privacy-Preserving Anonymization of Set-Valued Data," in *Proc. of Very Large Data Bases*, 2008.

[40] D. Christin, A. Reinhardt, S. Kanhere, and M. Hollick, "A Survey on Privacy in Mobile Participatory Sensing Applications," *Journal of Systems and Software*, vol.-84, no.-11, pp. 1928–1946, 2011.

[41] D. Christin, C. Rokopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, "Incognisense: An Anonymity-preserving Reputation Framework for Participatory Sensing Applications," *Pervasive and mobile Computing*, vol.-9, no.-3, pp. 353–371, 2013.

[42] D. Christin, J. Guillemet, A. Reinhardt, M. Hollick, and S. S. Kanhere, "Privacy-preserving Collaborative Path Hiding for Participatory Sensing Applications," in *Proc. of IEEE MASS*, 2011.

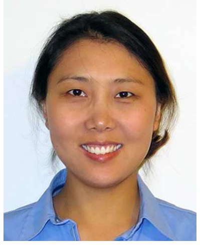
[43] E. De Cristofaro and C. Soriente, "PEPSI: Privacy-Enhanced Participatory Sensing Infrastructure," in *Proc. of ACM WISEC*, 2011.

[44] M. Terrovitis and N. Mamoulis, " Privacy Preservation in the Publication of Trajectories," in *IEEE Mobile Data Management*,2008.

[45] M. Gruteser and X. Liu , "Protecting Privacy in Continuous Location-tracking Applications," *IEEE Security & Privacy*, vol.-2, no.-2, pp. 28–34, 2004.

[46] X. Chen, X. Wu, X. Li, Y. He, and Y. Liu, " Privacy-preserving high-quality map generation with participatory sensing," in *Proc. of IEEE INFOCOM*, 2014.

Xi Chen received her Bachelor of Engineering degree and Bachelor of Economics degree from Xiamen University, P.R. China, both in 2011, and Master of Engineering degree in Tsinghua University, P.R. China, in 2014. She is now research assistant in Information Science and Technology National Lab, IoT Technology Center, THU, in Wuxi, Jiangsu Province. Her research interests include participatory sensing, mobile computing and privacy protection.



Xiaopei Wu received the B.Sc. degree in computer science from the Luoyang Normal University, China, in 2005 and the M.Sc. degree and Ph.D. degree in computer science from the University of Electronic Science and Technology of China in 2007 and 2012, respectively. She worked in the University of Michigan as a visiting researcher from 2008 to 2011 and joined the Tsinghua University, China, as a postdoctoral research fellow in October 2012. Her research interests include performance modeling, analysis, resource allocation issues in wireless sensor networks.

Dr. Xiang-Yang Li is a professor at the Illinois Institute of Technology. He is an IEEE Fellow and an ACM Distinguished Scientist. He holds EMC-Endowed Visiting Chair Professorship at Tsinghua University. He is a recipient of China NSF Outstanding Overseas Young Researcher (B). Dr. Li received MS (2000) and PhD (2001) degree at Department of Computer Science from University of Illinois at Urbana-Champaign, a Bachelor degree at Department of Computer Science and a Bachelor degree at Department of Business Management from Tsinghua University, P.R. China, both in 1995. His research interests include wireless networking, mobile computing, security and privacy, cyber physical systems, smart grid, social networking, and algorithms. He and his students won four best paper awards, one best demo award and was nominated for best paper awards twice (ACM MobiCom 2008, ACM MobiCom 2005). He published a monograph "Wireless Ad Hoc and Sensor Networks: Theory and Applications". IEEE Member 2000, SM 2008, Fellow 2015, ACM member 2000, Distinguished Scientist 2014.

Xiaoyu Ji is currently a fourth-year Ph.D. student in the department of Computer Science of Hong Kong University of Science and Technology. He received his Bachelor's degree in Electronic Information & Technology and Instrumentation Science from Zhejiang University, Hangzhou, China, in 2010. His research interests include protocol design in wireless ad-hoc and sensor networks, by exploiting the rich physical layer information. He is a member of IEEE.

Yuan He is an associate professor in the School of Software and TNLIST of Tsinghua University. He received his BE degree in University of Science and Technology of China, his ME degree in Institute of Software, Chinese Academy of Sciences, and his PhD degree in Hong Kong University of Science and Technology. His research interests include Internet of Things, sensor networks, pervasive computing, and cloud computing. He is a member of IEEE and ACM.

Yunhao Liu received his BS degree in Automation Department from Tsinghua University in 1995, and an MS and a Ph.D. degree in Computer Science and Engineering at Michigan State University in 2003 and 2004, respectively. Yunhao is now Chang Jiang Chair Professor and Dean of School of Software at Tsinghua University, China. Yunhao is ACM Distinguished Speaker, and now serves as the Chair of ACM China Council and also the Associate Editor for IEEE/ACM Transactions on Networking and ACM Transactions on Sensor Network. He is a Fellow of IEEE.