Privacy-Preserving High-Quality Map Generation with Participatory Sensing

Xi Chen*, Xiaopei Wu*, Xiang-Yang Li†, Yuan He*, Yunhao Liu*
*School of Software, TNLIST, Tsinghua University
†Department of Computer Science, Illinois Institute of Technology

***Abstract*—Accurate maps are increasingly important with the growth of smart phones and the development of location-based services. Several novel crowdsourcing based map generation protocols have been proposed that rely on volunteers to provide their traces. Although creative, those methods pose a significant threat to the users' privacy as traces imply user behavior patterns. On the flip side, crowdsourcing-based map generation method does need individual locations. To address this debacle, in this work, we present a systematic participatory-sensing-based high-quality map generation scheme, PMG, that meets the privacy demand of individual users. In our approach, individual users merely need to upload unorganized sparse location points which greatly reduces the risk of exposing privacy, and the server generates accurate maps with unorganized points instead of users' traces. Experiments show that our solution is able to generate a high-quality map for a real environment that is robust to noisy data. The difference between the ground-truth map and the map produced by this design is $< 10m$, even when the collected locations are about $32m$ apart after clustering for the purpose of removing noise.**

Keywords: privacy-preserving, map generation, *crust*

I. Introduction

During the last decade, portable smart devices have significant improvements in computing performance, memory size and the number of sensors embedded in them (*e.g.*, GPS, accelerometer, gyroscope). These improvements allow the devices to be adopted in more areas such as navigation, location-based services, social networking and etc. Most of those applications jointly exploit the integrated maps and user's current location to provide various kinds of service to users. Therefore, it is fundamental and indispensable to provide accurate and most-updated maps. Currently, digital maps (*e.g.*, Google Maps), based on the culmination of satellite imagery as well as street level information, are widely used. They, however, can not precisely reflect the most up-to-date map information, especially in the developing countries, where cities are often undergoing constructions, renovations and renewals, resulting in the integrated map 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 attentions. In those schemes, individual volunteers contribute their trace information (with GPS data) to a central map generation server. Despite of guaranteeing high quality map information, the methods often have various limitations, like energy inefficiency, and privacy leakage. As most existing methods exploit the traces contributed by each individual user, it raises a great concern of leaking the user's privacy. In this paper, we present a privacy preserving map generation scheme, PMG. To protect the user's privacy, unlike previous methods, in our scheme, each user selectively chooses, reshuffle, and upload a few locations from their traces, instead of the entire trace. After receiving these *unorganized* points cloud from a group of users, the map generation server will generate the final map.

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

Indeed reporting traces is not a good choice for protecting user's privacy. In PMG, we let each individual volunteer select a subset of points from his traces with respect to the privacy protecting demand. In our scheme, a user protects his privacy from two aspects. The first is to break the temporal relationship among reported points. Observe that the temporal relationship could be potentially exploited by the map generator to recover user behavior patterns, *e.g.*, Maskit [1] uses the Hidden Markov Model to recover user patterns. In our scheme, the volunteer shuffles the points from his trace and then report the shuffled partial collection to the map generation server, for obscuring the temporal relationship between 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 points a user should select and report. In this work we focus on the trace privacy: we say a trace is secure against the server if the server cannot (uniquely or approximately) recover the trace from reported points. We propose a mathematical formula to quantify the relationship between the number of reported locations and the degree of privacy leakage. We leave the protection of other privacy as a future work.

As to the server, the fundamental task is to reconstruct the underlying map from a group of unorganized location points. Clearly, we can't rely on the traditionally used trace-based map generation method (*e.g.*, CrowdAtlas [2]) that sequentially connects the points according to the sampled time label, since two adjacent points might not be consecutive in any trace. Thus, under the framework of privacy-preserving, it is not a trivial task to seek for an effective map generation algorithm with theoretical performance guarantee. In this work, we address the challenge of building a high-quality map from a set of unorganized points by borrowing theoretically sound *curve reconstruction* techniques from computational geometry. Based on this, we can show that when the sampling (the set of points reported by all volunteers) reaches a certain threshold, the quality of the generated map is assured. Here the sampling density required for high-quality map generation depends on the medial axis and local feature size, two inherent features of the map to be generated, We quantify the quality of the generated map by leveraging the Lower Bound of Voronoi

Angle (LBVA), a quantified metric describing the distance between the estimated map and ground truth [3].

The third challenge is to design map generation algorithm that is robust to the noisy data. Note that the collected GPS locations are not error free but unreliable GPS [4][5]. Typically, the GPS data will have an error at least $10m$. The sparsity of the sampled locations (for the purpose of privacy-protection), the small local feature size at some portion of the map, and the GPS error, will lead to inaccurate or even erroneous generated map. We first apply a simple GPS data filtering procedure to remove all potential unreliable data. By requesting sufficiently dense samples and carefully clustering the reported sampling points, we are able to show that our scheme is robust to GPS errors.

There are also many subtle details 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 constraint. 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 formulate our problem as the classical location point selection with the goal of maximizing the LBVA criterion, meanwhile satisfying the minimal requirement of privacy protection. We tackle the problem of maximizing LBVA (this will be referred to as MaxLBVA) for optimizing locations selection with privacy-preserving consideration. Due to the intractability of directly solving the MaxLBVA over unknown/infinite locations set (no prior knowledge abut the roads distribution, any point in the space could be candidate), we alternatively propose an equivalent MaxLBVA over a group of geography cells. We exploit the submodularity of the objective function (*e.g.*, LBVA) to develop an efficient approximation greedy algorithm, which achieves at least a constant fraction of the optimal solution.

We extensively evaluate this design 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 noise 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 II 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 III. We present our evaluation results in Section IV, review the related work in Section V and conclude the paper in Section VI.

II. Problem Formulation and Background

A. Problem Formulation

We assume that the participatory-sensing based map generation service is composed of one central data processing center and a group of volunteers spread over a geographic region. The center is in charge of collecting data (submitted voluntarily by these users or queried by the center) and producing a high-quality map from the set of locations collected (and possibly a prior knowledge of the map). We omit the incentive issues in this work as this is not the main scope of this paper. For the map generation, we do not assume that the center has a prior knowledge on 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 locations trace using smartphones. Each user will (voluntarily or based on the incentives provided by the center) provide some transformed data of the traces to the center for map reconstruction.

In this work, a map is mathematically defined as a geometric graph $G = (V, E)$ where V are 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 individual location information disclosure. To eliminate the possible risk of privacy exposure, one natural way is to let the user report fewer locations information. However, this will inevitably affect the quality of map generation service. To address the debacle between map quality and user privacy, in this work we will let each user update a subset of location points (which are randomly shuffled to remove the temporal ordering of points in the trace) to the server so as to minimize a certain measure of map generation error. This approach can assure that some constraints on individual location privacy exposure are satisfied. If not specified otherwise, throughout this paper the privacy we want to protect is the private location trace associated with each user.

More formally, consider m users and let $U_i (1 \leq i \leq m)$ be the set of collected GPS location 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:

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

where $\zeta(\cdot)$ returns the estimated map given reported GPS location points sets from m users, Err() is a certain error function measuring the distance between the real map $\mathcal{F}$ and the estimated, $PE_i(\cdot)$ 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.

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 points collected. We later will show that, if the set of collected points meets certain sampling quality condition, the reconstructed 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 of interest as one polygonal curve in 2D plane and the location points 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

[3] in computational geometry, which allow one to uniquely determine the original curve from a subset of samplings that satisfies some special condition. 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 (or lack of) of the data submitted by each user. A simple measure would be the number of points reported by the user: more points mean worse privacy protection. So a user may put a limit on the number of points reported, thus $\text{PE}_i(P_i)$ is simply the cardinality of set P_i. Obviously, report a large volume of 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, attacker 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 trajectory information of each user, we will introduce a novel privacy quantification scheme later in Section III-B. Intuitively, our privacy quantification assures that the attacker cannot recover the users' trace when certain conditions are met. Note that different privacy quantification functions could be integrated into our scheme, *e.g.*, we can define a sophisticated privacy leakage quantification based on the HMM model used in [1].

B. Curve Reconstruction

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

Consider a 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 ranging over minimum spanning tree[6], r-regular shapes[7] to α-shapes[8] have been proposed to find the solution of such problem, among which [3] shows one geometric graph, *Crust*, 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 estimate 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 corresponding endpoints belong to S.

Due to the existence of advanced and elegant program for Delaunay triangulation and Voronoi diagram[9], [10], computing *Crust* of one given finite set S is simple, easy to implement and scalable to the cardinality of S. Most 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 some relative definitions in [3] at first.

Definition 1. *[3] The* Medial Axis *of a curve $\mathcal{F}$ is closure of the set of points which have two or more closet points in $\mathcal{F}$.*

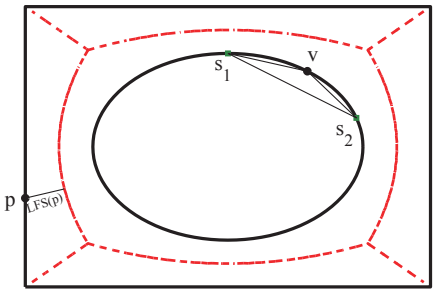


Fig. 1: Medial axis (in red), LFS(p), and Voronoi Angle.

Definition 2. *[3] 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. *[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 $\frac{D(p,s)}{LFS(p)} \leq \gamma$, where $D(p,s)$ represents the distance between p and s.*

Definition 4. *[3] 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. 1 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. 1. The solid black curve represents the original smooth curve. And its corresponding medial axis is showed as the dashed red curve.

Armed with these definitions, we will give two useful theoretical analysis in [3], denoted as the following two lemmas.

LEMMA 1. *Let S be a γ-sample from a smooth curve $\mathcal{F}$. Then (i) if $\gamma \geq 1$, $\mathcal{F}$ is un-reconstructible; (ii) if $\gamma \leq 0.252$, 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, the curve reconstruction problem is unsolvable when $\gamma \geq 1$. In other words, there may not be a unique graph on S that connects every pair of samples adjacent along that graph. 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. 1) 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 it is also desirable to minimize the cost and privacy leakage caused by collecting more points.

C. Alternative Objectives and Challenges

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

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

where $\Gamma(\cdot)$ 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.

However, there are 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 location points 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.

A second challenge is that GPS location data is not error free. A user can simply suppress the data if the error is more than a predefined threshold. However, doing this might not completely remove all potential error. 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 [11] 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 volunteering users and 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 participant sensing context.

III. Proposed Solution

A. System Architecture

Fig. 2 shows the overall architecture of our solution. At the network level, the system consists of a number of volunteering users and a map generation server.

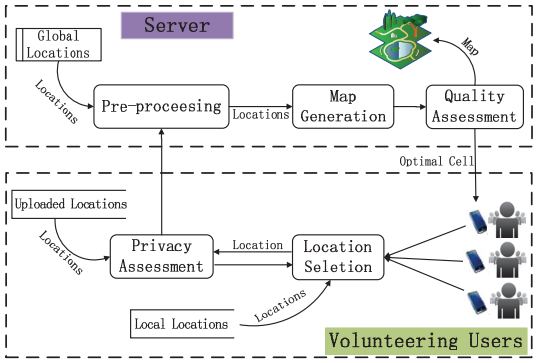


Fig. 2: The architecture of our scheme PMG

Volunteering Users: The volunteering 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 look-up 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 III-C. 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 volunteering users. To guarantee the estimated map quality, all chosen GPS locations will firstly enter into one data pre-processing block to remove all possible 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 Privacy 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 (*i.e.*, Location Selection) is further scheduled to estimate the optimal location points that will provide maximal gains in estimating the original map; server will broadcast these locations via *request* packet to *actively* pull the useful information. One practical optimal location selection will be introduced and analyzed in Section III-C.

B. Privacy-preserving Scheme

In this subsection, we develop one advanced and elegant privacy-preserving scheme based on the γ-sample condition, which actually corresponds to the constraint in **P**.

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 determine the real route that the user has been passing. Mathematically, this curve reconstruction problem is unsolvable. From Lemma 1, given a set of points that form a γ-sample, the bigger value γ doesn't bode well for the success of *Crust*, especially when $\gamma \geq 1$.

In order to protect the individual trace privacy, the user 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 γ-sample condition if adding the point to the set of historical reported locations within T. This new point is said to be *safe* (or *qualified*) if $\gamma \geq \gamma^{'}$, where $\gamma^{'}$ is the user-defined privacy requirement.

C. Near-optimal Location 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 submodularity. One simple greedy algorithm within constant ($\approx 63\%$) of the optimum is proposed.

1) Proposed alternative formulation: Obviously, MaxLBVA remains the combinatorial optimization which is intractable. At the very beginning, however, from sporadic collected locations, the server could roughly infer that there might be several roads existing in one physical region. Therefore, the server might benefit most by choosing locations within such region. To bridge the gap between server's oblivious to the set of locations currently collected by all users, and server's requiring the set of candidate locations to improve map quality, we will partition the region into a group of cells. Given historical knowledge and collected locations, the server will estimate the 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 ask 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

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

where $\mathbb{E}[\cdot]$ denotes expectation operation, computed over all locations uniformly distributed within the cells.

The function $\Gamma(\cdot)$ we defined is over the locations 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' movement pattern, it is reasonable to assume that 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

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

where r is the side length of a cell, p is a location in the cell and S_i is the locations collected by the server after choosing the first i cells. Here S_0 means the initial sporadic collected locations by the server. 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.

2) Properties of the objective: There are three important 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. The proof can be found in our online technical report[12].

Theorem 1. *Consider a smooth curve $\mathcal{F}$. Let V 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.

3) Proposed greedy algorithm: In general, maximizing submodular function is NP-hard [11]. We instead use a heuristic greedy algorithm to approximate the optimum. It operates as follows: starting from $A = \emptyset$, iteratively adds 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, once the server computes the optimal cell so far through Eq. (1) and (2). Then it broadcasts a *query packet* containing the physical information (*e.g.*, GPS coordinate for the cell's four corner points) of the chosen cell. Any user hearing such query packet will examine whether their stored location points falling in such cell. If the matched location point set is nonempty, the privacy-preserving scheme is further applied on them to remove all non-safe data that might lead to private trace leakage. In response to the *request* packet, eventually, the user will send the final chosen safe location points back to the server.

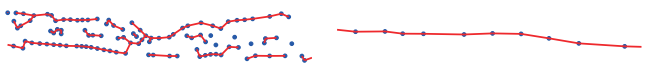
We end this part by discussing the theoretical bound of our proposed simple greedy algorithm. Since our quantified objective is submodular, non-decreasing and with $\mathcal{R}(\emptyset) = 0$, the below theorem turns out that our algorithm could achieve a constant-factor ratio to the optimum.

Theorem 2. *[11] Let $\hat{A}$ be the chosen cells by the greedy algorithm and $A^* = \max_{A \subset V} \Gamma(A)$. Then*

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

D. Impact of GPS Sample Error and Road Width

Two critical issues must be addressed to make our protocol practical: 1) GPS sample error (thus, samples 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 with zero thickness, and the unorganized points precisely from the underlying curve. While for 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



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

Fig. 3: Impact of Road Width.

add extensive unnecessary roads/edges within the same road, especially when the road width is very large (*e.g.* high way). Fig. 3 illustrate an example.

We address these challenges using a one-stone-two-birds solution: down-sample the reported data from users. For dealing with GPS error, we first remove the data when such accuracy is more than a threshold η. Even so, note that the uploaded data is still noisy. For the goal of map generation, it is reasonable to set η to be on the same order of the road width. We then apply a simple clustering algorithm to the filtered GPS data. To be specific, 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 data 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 as the increase of the number of reported users, the empirical mean will be close to the real value with higher probability (close to 1). From the Hoeffding's inequality, we have

Theorem 3. *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*

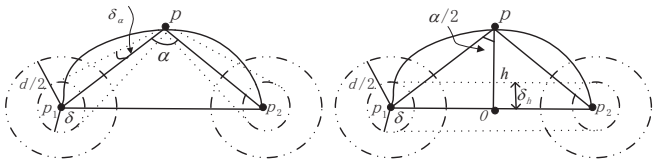
$$Pr(|\frac{1}{n}\sum_{i=1}^{n} x_i - x_0| \geq \delta) \quad \leq \quad 2exp(-n^{\frac{1}{3}}), \tag{3}$$

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

Using the collected noisy GPS data, we examine the performance of the Voronoi Angle (*i.e.*, α in Fig. 4(a)) and the maximal Euclidean distance between the real curve and the estimated (*i.e.* h in Fig. 4(b)) using down-sampled data. Consider two consecutive sample points p_1 and p_2 on a smooth curve $\mathcal{F}$, as shown by Fig. 4. 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.

From Theorem 3, 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 geography knowledge, the value of $\delta_\alpha = \arcsin\frac{\delta\gamma}{\tan(\arcsin\frac{\gamma}{2})Ds(p_1,p_2)}$ 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



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

Fig. 4: The effect of GPS data error.

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).

IV. Performance Evaluation

We present in this section 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 III-C to choose the optimal locations. The map generator we use is *Crust*.

We will use two data sets. The first one, also referred to as the *Shanghai Data*, is a group of GPS datasets published on the CrowdAtlas website with 24 traces containing 954000 locations in total [13]. The area of this data set is about $149.09km^2$ and the total length of tracks is $111390m$. A second data set, also referred to as the *Wuxi Data*, was collected at the Wuxi New district, in total 323120 locations. And its total area and tracks are $36.45km^2$ and $29284m$, respectively.

Due to the lack of large scale participant sensing filed, we reshuffle the two data sets and randomly assign these data points into m different files to emulate the number of volunteering 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).

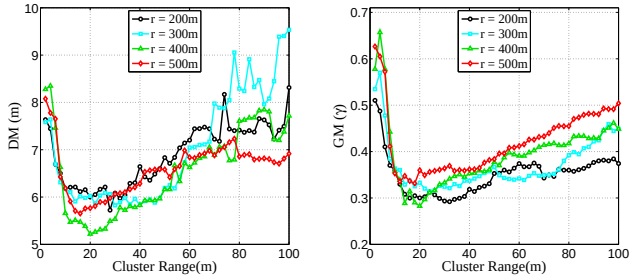
Denote the recovered segments set as $\hat{E} = \{e_i, 1 \leq i \leq |\hat{E}|\}$, each segment with n^i points. We next will use two metrics to verify the effectiveness of PMG: a first one is Deviation Metric(DM) denoting how far is the estimated map from the ground truth, and the other is Gamma Metric(GM),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

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

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.

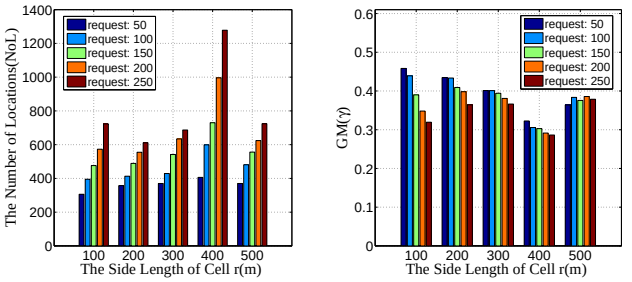
A. Impact of different parameters

In this subsection, we observe the impact of different parameters (*i.e.* cluster range and cell size) on the final estimated map quality. We conducted our experiments on the



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

Fig. 5: The impact of cluster range.



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

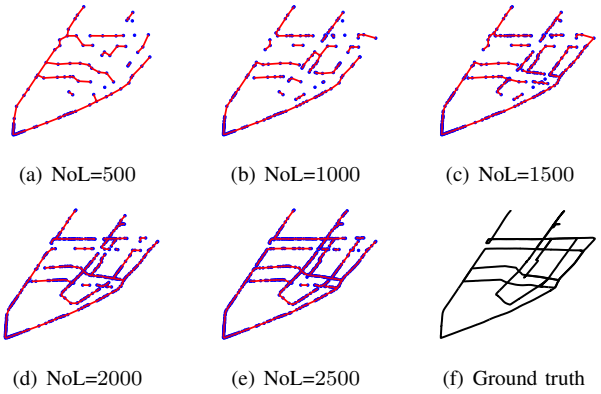
Fig. 6: The impact of cell size

two datasets, the effect of which share similar trend. Due to space limitation, we only report the results on *Wuxi data*, more information could be found in [12].

1) Cluster range: Fig. 5 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 length of cell (*i.e.* $r = 200, 300, 400, 500$, all in units of m).

Regardless of the cell size length 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 $30m$. In addition, the quality of the generated map could achieve the empirical optimum/minimum when the cluster range is around $20m$, which is physically consistent with the real road width (about $15m$ and $25m$) where we collected data. Note that since 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 $30m$.

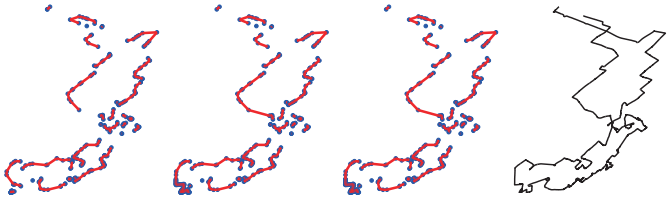
2) Cell size: We next examine the effect of cell size on the generated map quality. Due to the performance similarity between DM and GM, we only offer the performance of GM under different cell size. In addition, since our locations selection algorithm is cell-based, we also investigate the impact of 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 is shown by Fig. 6.



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

(d) NoL=2000 (e) NoL=2500 (f) Ground truth

Fig. 7: *Wuxi data*: generated maps at different NoL



(a) NoL=500 (b) NoL=1000 (c) NoL=5000 (d) Ground truth

Fig. 8: *Shanghai data*: generated maps at different NoL

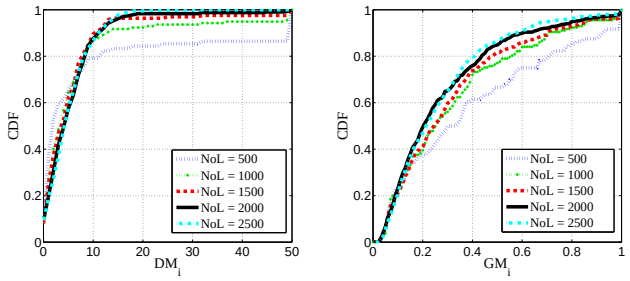
From Fig. 6(a), we can see that as the increase of r, NoL increases at first, achieves a peak when $r = 400$, then begins to decrease. And GM behaves the opposite trend. The result is reasonable. When cell size is small, each cell might contain a few matched locations, so after hearing the *request* packet, less users will response. As the increase of r, more qualified locations might be contained in each cell, lending to the increases of NoL. However, when $r = 500$, the cell size will be very large. It is possible that the chosen cell will contain many roadless areas, even such cell achieving the highest marginal gains. This suggests that $r = 400$ is the empirical optimal value.

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 that more *request* packets mean more locations will be collected, thus leading to the improvement of the final generated map quality (*i.e.* the decrease of GM). However, there is a small exception for GM when $r = 500$ (see, Fig. 6(b)). Again this is due to that r is too large, containing many areas without roads which might lead to significantly small qualified location.

B. The quality of generated map

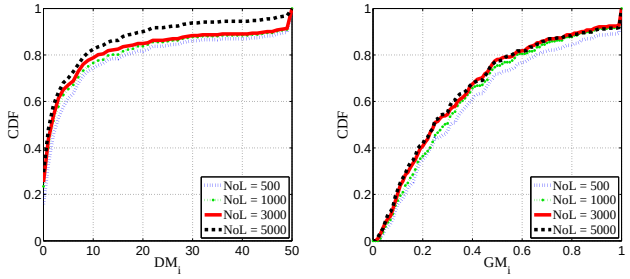
In this section, we will investigate the generated map quality in various dimensions. Unless otherwise stated, we will set the cluster range and size length of a cell r to be $20m$ and $400m$ in the next experiments.

1) Visual comparison: We first visually observe the generated map quality under different sampling points both in Wuxi and Shanghai, as shown by Fig. 7 and 8. Here the red lines mean the recovered segments; the blue points represent the clustered sampling locations. For comparison purpose, we also



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

Fig. 9: CDF observation with *Wuxi data*



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

Fig. 10: CDF observation with *Shanghai data*

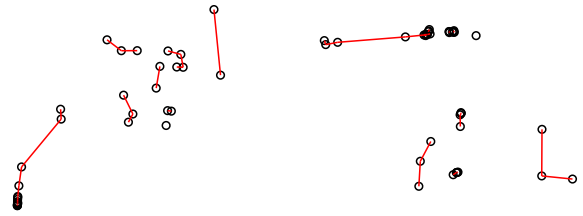
provide the ground truth in black, as shown by Fig. 7(f) and 8(d). As expected, the more of sampling locations, the better of recovered map quality. In Wuxi experiment, Fig. 7 shows that when NoL is equal to 2500, the recovered map could almost capture the general tread of the original map. For the Shanghai data, the performance improvement is very small if changing the number of locations from 500 to 1000. Moreover, such improvement will disappear when NoL > 1000.

2) Quantitative evaluation: To be more precise in quantitative comparison, we further observe the CDF of segment DM and GM under different number of sampling locations, as showed by Fig. 9 and 10. The results suggest that the estimated map based on Wuxi data outperforms than Shanghai data. For instance, when NoL=1500, about 90% of the recovered segments are at most 10m apart from the ground truth, while for Shanghai data, there are only 80% such segments even NoL=3000.

More statistical information describing the recovered map quality is presented in Table I. Here, NoLC means the number of locations after clustering and Density represents the average distance of consecutive clustered points. The recovered map quality will improve as the increase of sampling locations. Such improvement could also be verified by the increase of Density, decrease of DM and Variances. Compared with Wuxi data, the performance gains are not obvious for the Shanghai data, especially for NoLC. There are only 12 new added clustered points even if adjusting the NoL from 3000 to 5000. This is because that the number of volunteering 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 denser which results in the less NoLC and the slow growth of the quality. Therefore, it is highly necessary to

TABLE I: Map Generation Results with Two Data Sets

DataSet	NoL	NoLC	DM	Variance	$\lvert\hat{E}\rvert$	Length	Density
Wuxi	500	221	128.40m	421.18m	96	3776.9m	17.09
	1000	230	38.44m	205.94m	157	4941.1m	21.48
	1500	273	25.31m	178.13m	256	8153.9m	29.87
	2000	432	6.04m	7.83m	416	13430m	31.09
	2500	594	5.53m	4.26m	578	20186m	33.98
Shanghai	500	273	8.93m	15.92m	246	7041.3m	25.79
	1000	304	8.01m	15.22m	277	6424.2m	21.13
	2000	310	7.79m	15.05m	284	6335.6m	20.44
	3000	320	7.43m	14.86m	292	7100.6m	22.19
	4000	326	7.34m	14.73m	299	7893.0m	24.21
	5000	332	7.38m	14.6m	307	7945.6m	23.93



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

Fig. 11: Recovered individual trace by one user within one day

select appropriate parameters based on real situation, *e.g.* the number of users, the cell side length and so on.

C. Evaluation of Privacy Protection

We examine the performance of privacy protection by observing the individual recovered trace.

We set the privacy quality (*i.e.* $\gamma^{'}$) to be one. For simplicity purpose, each volunteering user exploits exactly same privacy protection quality. 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. Fig. 11 illustrates the recovered individual trace. Clearly, these two graphs contain many separated segments and points. This suggests that even if the server can effectively and accurately recover the unobserved map, it is impossible to infer each volunteering user's private trace.

V. Related Work

Map Generation: Nowadays, many mapping projects with the crowdsourcing activities have been successfully implemented, *e.g.* OpenStreetMap [14] and Google Map Maker [15]. Recently, Wang et al. develop an application named CrowdAtlas [2] to automate outdoor map update based on user traces, either individually or crowdsourced. Shen et al present Walkie-Markie [16] to generated indoor map based on user trajectories and use Wifi-Marks based on the RSS trend to locate. None of these protects user privacy and theoretically assures the map quality.

Curve Reconstruction: α-shape is one of the curve reconstruction methods to uniquely determine 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. γ-neighborhood graph [17] needs that the sampling density should be the same in each part of the curve. Another method β-Skeleton [18] is similar to γ-neighborhood graph except that the radius of the $forbidden\ region$ of two points in β-Skeleton is the same. Furthermore, it is like α-shape that β-Skeleton needs to choose an appropriate threshold β to ensure the results of curve reconstruction.

Privacy: Prior work has considered preserving privacy by four main strategies.

(1)*Restrict access:* Users can specify rules automatically to decide whether to release data [19][20], *e.g.* MaskIt [1] defines privacy with respect to sensitive contexts specified by users and can be preserved by filtering releasable context stream. However, malicious attackers can still intercept user data.

(2)*Fabricate data:* It makes users generate false data (*e.g.* dummies [21]) to send to service providers, *e.g.* PMP [22] protect privacy for iOS by providing a random or fake location to prevent profiling. However, it compromises data authenticity and could bring serious damage for some services.

(3)*Anonymous location information:* k-anonymity model [23] removes some features such that each item is not distinguishable among other k items, but its data have huge overhead. Mix zone model [24][25] assigns users in mix zones different pseudonyms to hide their paths, but it still permits the operation of many short term location aware. Furthermore, user behavior patterns are still predictable with low user density and most of them require a trusted middleware system.

(4)*Cryptography:* Secure multi-party computation [26], a subfield of cryptography, is to created methods that enables parties to jointly compute a function over their inputs, while at the same time keeping these inputs private. However, it consumes too many computing resources.

VI. 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*, we designed, implemented, and evaluated PMG for high-quality map generation with privacy-protection for each volunteer. Our scheme meets the individual users privacy demand and is robust to inherent noises of GPS data. The effectiveness of our proposed algorithms is validated through extensive numerical experiments over two real city-scale GPS data traces. 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 many future directions to pursue. One is, if we have certain prior knowledge about the road segments, whether it is possible to design more efficient algorithm to choose the locations with maximal gains in estimating the underlying map. We also would like to design schemes that can recover more detailed road conditions such as one-way road or two-way road, the traffic load distributions of different road segments.

References

[1] M. Götz, S. Nath, and J. Gehrke, "Maskit: Privately releasing user context streams for personalized mobile applications," in *Proc. of ACM SIGMOD on Management of Data*, 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] 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.

[4] S. Han, Q. Zhang, and H. Noh, "Applying filtering techniques to improve GPS positioning accuracy," in *ASAE Meeting Paper*, 2001.

[5] J.-y. Huang and C.-H. Tsai, "Improve gps positioning accuracy with context awareness," in *Proc. of IEEE Ubi-Media Computing*, 2008.

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

[7] D. Attali, "r-regular shape reconstruction from unorganized points," *Computational Geometry*, vol. 10, no. 4, pp. 239–247, 1998.

[8] F. Bernardini and C. L. Bajaj, "Sampling and reconstructing manifolds using alpha-shapes," in *In 9th Canadian Conference on Computational Geometry*, 1997.

[9] D. J. Mavriplis, "An advancing front delaunay triangulation algorithm designed for robustness," *Journal of Computational Physics*, vol. 117, no. 1, pp. 90–101, 1995.

[10] M. Iri, K. Murota, and T. Ohya, "A fast voronoi-diagram algorithm with applications to geographical optimization problems," in *System Modelling and Optimization*, 1984.

[11] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[12] X. Chen, X. Wu, Y. Li, Y. He, and H. Liu, "PMG: Privacy-preserving high-quality map generation with participatory sensing," http://www.cs.iit.edu/∼xli/paper/Conf/PrivateMap.pdf.

[13] "Crowdatlas website," http://grid.sjtu.edu.cn/mapupdate/.

[14] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[15] "Google map maker," http://www.google.com/mapmaker/pulse.

[16] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-Markie: indoor pathway mapping made easy," in *Proc. of USENIX NSDI*, 2013.

[17] R. C. Veltkamp, "The γ-neighborhood graph," *Computational Geometry*, vol. 1, no. 4, pp. 227–246, 1992.

[18] D. G. Kirkpatrick and J. D. Radke, *A Framework for Computational Morphology*. In G. T. Toussaint (editor), Computational Geometry, Elsevier Science Publishers, 1985.

[19] G. Myles, A. Friday, and N. Davies, "Preserving privacy in environments with location-based applications," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 56–64, 2003.

[20] U. Hengartner and P. Steenkiste, "Protecting access to people location information," in *Security in Pervasive Computing*, 2004.

[21] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proc. of ICPS*, 2005.

[22] Y. Agarwal, M. Hall, P. Gupta, L. Dolecek, N. Dutt, R. Gupta, R. Kumar, S. Mitra, A. Nilolau, T. Rosing *et al.*, "ProtectMyPrivacy: Detecting and mitigating privacy leaks on iOS devices using crowdsourcing," *Proc. of ACM MobiSys*, 2013.

[23] 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.

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

[25] ——, "Mix zones: User privacy in location-aware services," in *Proc. of IEEE Pervasive Computing and Communications Workshops*, 2004.

[26] 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.