RTS Assisted Mobile Localization: Mitigating Jigsaw Puzzle Problem of Fingerprint Space with Extra Mile

Chao Song*†‡, Jie Wu†, Li Lu*, and Ming Liu*‡
*School of Computer Science and Engineering, University of Electronic Science and Technology of China
†Department of Computer and Information Sciences, Temple University
‡Big Data Research Center, University of Electronic Science and Technology of China
Email: {chaosong, luli2009, csmliu}@uestc.edu.cn, {jiewu}@temple.edu

***Abstract*—With the development of Location Based Services (LBSs), both academic researchers and industries have paid more attention to GPS-less mobile localization on mobile phones. The majority of the existing localization approaches have utilized signal-fingerprint as a metric for location determinations. However, one of the most challenging issues is the problem of uncertain fingerprints for building the fingerprint map, termed as the *jigsaw puzzle problem*. In this paper, for more accurate fingerprints of the mobile localization, we investigate the changes of Received Signal Strength Indication (RSSI) from the connected cell-towers over time along the mobile users' trajectories, termed as RSSI Time Series (RTS). Thus, we propose an RTS Assisted Localization System (RALS), which is a GPS-less outdoor mobile localization system. For localization, an RTS map is built on the back-end server, which consists of RTS harvested from the mobile phones, by the way of crowdsensing. The jigsaw puzzle problem slows down the map construction solely by the regular unintentional users with short-distance trajectories, and affects its efficiency. To speed up the map construction, we propose employing a few advanced intentional users with additional long-distance trajectories, at a higher cost than the regular user; this is called *extra mile*. Our extensional experiments verify the effectiveness of our localization system.**

***Keywords*—Crowdsensing, jigsaw puzzle problem, outdoor mobile localization, RSSI time series**

I. Introduction

Over 6.8 billion mobile phones were in use all over the world in 2013 [1]. With an increase in the number of mobile applications, many outdoor mobile applications are getting a lot of attention from both academic researchers and industries [2], such as Waze, and pothole detection [3]. Global Positioning System (GPS) is the most popular technology for outdoor mobile localization, with high accuracy. However, it is well-known that localization with GPS has high energy consumption in mobile phones [4]. Second, GPS cannot receive the satellite signals in "urban canyons" (such as nearby tall buildings, or by surrounding tunnels) [5], [6]. Third, since not all mobile phones are equipped with GPS, it is required that the mobile users additionally take the GPS devices [5], [6].

Therefore, the GPS-less localization system is required for outdoor mobile applications, which has been investigated by both academic researchers and industries. The majority of the existing localization approaches have utilized signal-fingerprint as a metric for location determinations. Yang *et al.* in [7] propose to utilize RSS fingerprints, which can be the WiFi signal strengths from multiple access points at every location of an interested area, and accordingly build a fingerprint database (a.k.a. radio map) in which fingerprints are associated with the locations where they are recorded. However, for the outdoor mobile localization, it is difficult to record and process the fingerprints of all the positions along the trajectory, due to the large amount of fingerprints. Thus, the *trajectory-fingerprint-based approach* has been proposed for the mobile localization. The phones carried by the mobile users autonomously and continuously record the ambient conditions, and position information as the fingerprint of the user's trajectory [4], [8]. Paek *et al.* in [9] propose to utilize a cell-ID sequence-based localization, which is recorded by changes of the IDs of the connected cell-towers by the mobile phone along the user's trajectory. Generally, the mobile phone captures the signals of multiple cell-towers at one place, and connects to the cell-tower with the strongest signal strength [10]. Thus, the existence of the RSSI fluctuation problem can cause changes to the connected cell-tower, even at the same place, and also reduces the accuracy of the cell-ID sequence-based localization.

However, signal-fingerprint-based approaches require a lengthy and extensive training phase to construct a signal fingerprinting database or map, which can be constructed by way of two different approaches. Traditional methods of building a fingerprint map involve a site survey process (a.k.a. calibration), in which engineers record the RSS fingerprints (e.g., WiFi signal strengths from multiple access points) at every location of an interested area, and accordingly build a fingerprint database in which fingerprints are related with the locations where they are recorded [7]. However, site survey is time-consuming, labor-intensive, and vulnerable to environmental dynamics. To reduce the cost of map construction, another approach without site survey has been proposed [7], [10], by the way of crowdsensing with some *regular unintentional users*. Due to the well-known signal fluctuation problem, the signal-fingerprints at faraway positions may be similar [11], which cause errors within map construction. Thus, we investigate the problem of the uncertain signal-fingerprints in the map construction, termed as the *jigsaw puzzle problem*, which requires a large number of the regular unintentional users. As a result, the problem slows down the map construction, and severely affects its efficiency.

In this paper, for a highly accurate trajectory-fingerprint in outdoor mobile localization, we investigate the changes of RSSI of the mobile phone from the currently connected cell-tower, which is termed as *RSSI Time Series* (RTS). Thus, we propose an energy-efficient, and highly accurate localization system for mobile users, namely RTS Assisted Localization System (RALS), by the way of crowd-participatory sensing.

The requirement for the phone configuration is that the phone can provide the information of the connected cell-tower including the RSSI and the cell-ID, which are supported by most of the mobile phones. Here, the cell-ID is used to narrow the searching range. A mobile user can query the location from the server, by sending the RTS along his traveled trajectory. The server matches the received RTS with the fingerprint database (a.k.a. RTS map) for searching the continuous road segments with the similar RTS. Thus, the server can track the road segment where the querying user currently moves. In addition, with the help of the peaks along the RTS on the current road segment, the server can locate the user's position, and feeds it back to the querying user.

To speed up the map construction with the jigsaw puzzle problem, we propose a novel crowdsensing-based scheme by a few *advanced intentional users*. The advanced intentional user additionally moves a longer trajectory with non-overlapped positions, termed as *extra mile*, but with a higher cost than a regular user. The basic idea behind this scheme is that, a long-distance trajectory without overlapped road segments has more contributions to speed up the map construction, because it has a higher probability of distinguishing the uncertain signal-fingerprints in the RTS map. Many studies propose that the mobile nodes with social characteristics generally visit community homes frequently, while other locations are visited less frequently [12]. For example, in a campus, the students always move from one building to another, such as from a dormitory, a teaching building, or a restaurant. Although the costs of harvesting the fingerprints from regular unintentional users are cheap, their trajectories are generally short, which make it difficult to distinguish the uncertain RSSI-fingerprints, and cause the jigsaw puzzle problem. We discuss the efficiency of the crowdsensing-based scheme, by considering the cost of the intentional users, and comparing the sequential and parallel patterns for employing the intentional users.

We conduct a comprehensive investigation of the GPS-less mobile localization, and our technical contributions are multi-fold, including: (1) We propose an RTS assisted localization system, namely RALS, which utilizes the RSSI time series for energy-efficient mobile localization. RALS utilizes the similarity of RTS on the same road, and the peaks along the RTS. The system is based on the way of crowdsensing by the regular unintentional mobile users. (2) To speed up the map construction with the jigsaw puzzle problem, we propose a novel crowdsensing-based scheme called extra mile, combined with the regular unintentional users, and a few advanced intentional users. (3) Our extensional real experiments verify the effectiveness of RALS and our crowdsensing-based scheme.

The remainder of this paper is organized as follows: Section II introduces the RTS assisted localization system, based on our observations with real experiments; in Section III, we present the crowdsensing-based scheme of map construction; Section IV evaluates the performance of the proposed system; in Section V, we survey the related work; the last section concludes this paper and presents our future work.

II. RTS Assisted Mobile Localization

In this section, we first discuss the existing solutions for the energy-efficient mobile localization, with some observations from our real experiments. Then, we introduce the RTS assisted mobile localization. At last, we introduce the architecture of our system RALS.

(a) (b)

Fig. 1. The connected cell-towers along the same route

A. Energy-efficient Mobile Localization

The traditional celltower-based localization is highly energy-efficient, but it incurs errors as high as 500 meters, which may be insufficient for the outdoor mobile localizations [9]. Furthermore, many studies propose some energy-efficient mobile localization schemes without the road map, for the applications based on the arrival time prediction [9], [10], because the travel time under them is more important than the physical position [13]. Paek *et al.* in [9] present a cell-ID sequence as the trajectory-fingerprint for outdoor mobile localization. However, due to the RSS fluctuation problem, the connected cell-tower at the same place can be changed. Thus, on the same path, the cell-ID sequence recorded by the mobile node can also be different, as can the cell-ID transition points, which have been discussed in [9], [10].

Our real experiment is as follows: we drive a car along the same path several times in Chengdu, China. The mobile phone records the information of the connected cell-towers, which includes the ID and position (i.e. longitude and latitude). The interval time of two consequential records is less than one hour. Figure 1 shows the two cell-ID sequences on Google Map. We notice that the connected cell-towers along the two trajectories are different. Moreover, the cell-ID transition points are also changed. Zhou *et al.* in [10] propose a top-k cell-tower set sequence matching method to classify the reported cell-tower sequences, which requires to record the information of neighboring cell-towers. However, many mobile phones can only provide the information of currently connected cell-towers, and cannot provide the information of neighboring cell-towers.

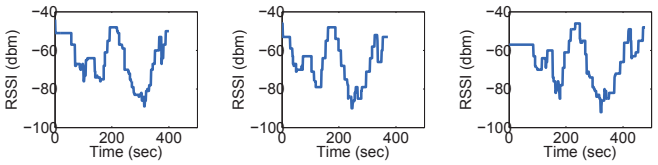
B. RSSI Time Series

For more accurate fingerprints of mobile localization, we investigate the RSSI time series (RTS) from the connected cell-towers along the mobile users' trajectories. An RTS is a sequence of data points, typically measured at successive points in time spaced at uniform time intervals, as follows:

$$rts = \{(t_1, rssi_1), (t_2, rssi_2), \cdots, (t_n, rssi_w)\},$$

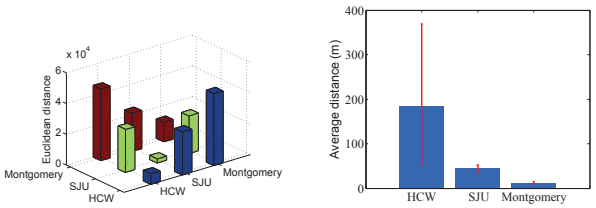
where t_i denotes the time index, $rssi_i$ denotes the RSSI at each time slot, and w denotes the real-valued observations.

In our real experiments, three drivers drove their cars at different speeds, on the highway from Chengdu to Wenjiang



(a) 62 km/h (b) 67 km/h (c) 52 km/h

Fig. 2. Similar RTS on the same road with different average moving speeds



(a) Similarity (b) Physical distance of peaks

Fig. 3. Compare the RTS from different places

in China, where their mobile phones in their cars recorded the RTS as shown in Figure 2. The RTS in Figure 2(b) is gathered two days after Figure 2(a), and the RTS in Figure 2(c) is gathered three days later after Figure 2(a). Due to the different speeds, the lengths of the RTS are different. Intuitively, we notice that the shapes of the three RTS in Figure 2 are similar. Like many studies on similarity [7], we utilize the Euclidean distance for measuring the similarity between two RTS, and a small distance between two RTS means they are more similar. However, the speeds of the mobile users on the same road can be different, which cause different lengths of RTS. For the alignment of RTS, we adopt the algorithm of dynamic time warping (DTW). Dynamic Time Warping is a classic dynamic programming algorithm which has been widely used for alignment and measuring similarity between two temporal sequences, which may vary in time or speed [14].

We did the same type of experiment with different drivers in three different places, 10 RTS from each place. Figure 3(a) shows the average pairwise distances of RTS from these places. The three places are as follows: (1) HCW denotes a highway from Chengdu to Wengjiang in China, with an average moving speed of 15.5 m/s; (2) SJU denotes the route of the campus bus in Shanghai Jiaotong University in China, with an average moving speed of 5.9 m/s; (3) Montgomery denotes West Montgomery Avenue on the main campus of Temple University in USA, with an average moving speed of 1.3 m/s. We notice that the RTS from the same road are much more similar than those from different roads.

C. Peaks of RTS

For indoor localization, some studies [15], [16] propose to utilize the trend of the received WiFi signal strength changes from increasing to decreasing when moving along the pathway. To mitigating the RSS fluctuation problem, we investigate the peaks along the RTS from connected cell-towers for outdoor localization, as shown in Figure 4(a). The peaks of the RTS are defined as the sets of the continuous data points with local maximum RSSI, in a section of RTS with the trend of RSSI changing from increasing to decreasing, searched by a sliding window. Figure 3(b) shows the average pairwise physical distance of the aligned peaks from every two places or the same place. The distance errors of different places vary, due to the different moving speeds. The average distance errors are less than 200 meters, which is much less than the traditional celltower-based localization mentioned above. Especially, the average distance error on Montgomery is the smallest, which is less than 12 meters. Thus, the peaks of the RTS along the same path appear to be the same positions.

For illustrating the reason for using the peaks of RTS to mitigate the RSS fluctuation problem, Figure 4(b) shows an example in which a vehicle first travels through the coverage of a cell-tower, and then travels through an overlapped area between the coverage of two cell-towers. A, B and C are the transition points of the connected cell-towers in its trace. For the coverage of the single cell-tower, the signal strength at the middle point M_1 can be larger than those at A or B, since M_1 is the nearest position to the cell-tower. Generally, the mobile phone captures the signals of multiple cell-towers at one place, and connects to the cell-tower with the strongest signal strength [10]. For the overlapped area between the coverage of two cell-towers, the connected cell-ID at the middle point M_2 may be changed, due to the existence of the RSSI fluctuation problem. However, the signal strength at the middle point M_2 is also the highest, relatively speaking. Thus, RALS utilizes the peaks of RTS as landmarks for localization, which is introduced in the next section.

D. System Overview

Based on the observation and the analysis of RTS, we propose the RTS Assisted Localization System (RALS). Figure 5 shows the architecture of RALS. RALS includes two major components, which are the mobile users and the back-end server. The mobile users have two responsibilities: (1) *probing user*, who uses mobile phones as well as the sensors to sense and report the RTS to the server; (2) *querying user*, who queries the location and LBS information (such as alert information). The *back-end server* is responsible for collecting the RTS from the probing vehicles, and intellectually processing such information so as to build the RTS map.

The system of RALS includes two phases, which are the *training* phase, and the *locating* phase. The training phase is completed by both of the probing users and the back-end server by the way of crowd-participatory sensing. The probing users gather the RTS while they are moving. Then, they upload the recorded RTS to the back-end server, when there is an available connection to the Internet, such as WiFi and 3G/4G. In the locating phase, the querying user also gathers the RTS where it has moved, and then uploads it with the locating query to the back-end server. The back-end server matches the uploaded RTS with the RTS map, and tracks the road segment where the querying user currently moves by the matchable RTS. Thus, with the help of the peaks, the server can locate the position of the querying user at the RTS map, and also predicts the arrival time of the alert information for the querying user.

Before uploading the RTS to the back-end server, the mobile phone partitions the RTS into several segments with the markable intersections. Many approaches can be used for sensing the intersections. One available approach is to

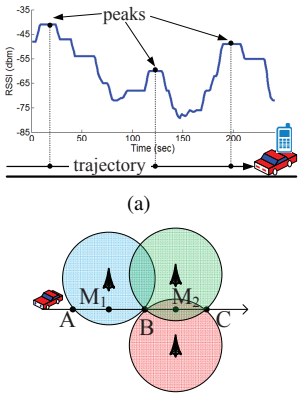


Fig. 4. Peak of RTS

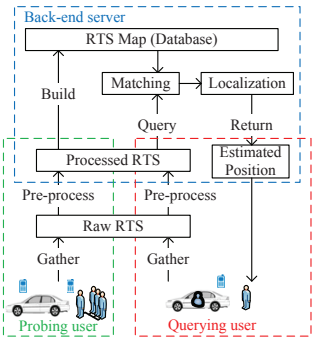


Fig. 5. System architecture

sense the features of surround by the mobile phones, such as the traffic lights at the intersections [17], or the ambience fingerprinting [18]. Another available approach is to sense the motions of the mobile users by the acceleration sensors in the mobile phones, such as in a stop-and-go pattern [19], [20], or the way of dead-reckoning [7], [21]. For the noises from the raw RTS, the mobile phone filters the RTS by the median filter algorithm (a classic filtering method), due to its computational efficiency. In order to reduce the overhead of data transmission, the uploaded RTS is compressed by the Shannon sampling theorem with a pre-defined sampling frequency. By comparing the similarities among the uploaded RTS, the back-end server merges the similar RTS as the same overlapped road, and connects the road segments by the order along the trajectories in the RTS map (fingerprint database).

E. Matching RTS of the Road Segment

To track the current road where the querying user is moving, the back-end server matches the uploaded RTS from the querying user with the RTS map in the locating phase. The matching process includes two steps: first, to narrow the searching range, the server selects the RTS of the road segments in the RTS map which has partially similar cell-IDs of the uploaded RTS to match; second, the server calculates the Euclidean distance between the uploaded RTS and the selected RTS, which are aligned by DTW. Generally, for two aligned RTS of road segments $rts_i = \{rssi_1^{(i)}, rssi_2^{(i)}, \cdots, rssi_w^{(i)}\}$ and $rts_j = \{rssi_1^{(j)}, rssi_2^{(j)}, \cdots, rssi_w^{(j)}\}$, define their difference (or dissimilarity) δ_{ij} by the Euclidean distance as follows:

$$\delta_{ij} = \left\| rts_i - rts_j \right\| = \sqrt{\sum_{h=1}^{w} (rssi_h^{(i)} - rssi_h^{(j)})^2}$$

If the dissimilarity δ_{ij} is smaller than a threshold ε, then they are treated as the same road segment. Otherwise, if $\delta_{ij} > \varepsilon$, rts_i and rts_j are treated as two different road segments. The determination of ε is based on the fingerprint samples collected at a given location, as introduced in [7].

A querying user sends a query for localization to the server, which includes the RTS and connected cell-ID sequence along the trajectory where the user has moved. The server matches the uploaded RTS with the RTS map in the database, for tracking the roads where the querying user has moved. Obviously, the longer length of RTS can increase the accuracy of matching, which will be discussed in Section IV.

F. Localization on the Matchable Road Segment

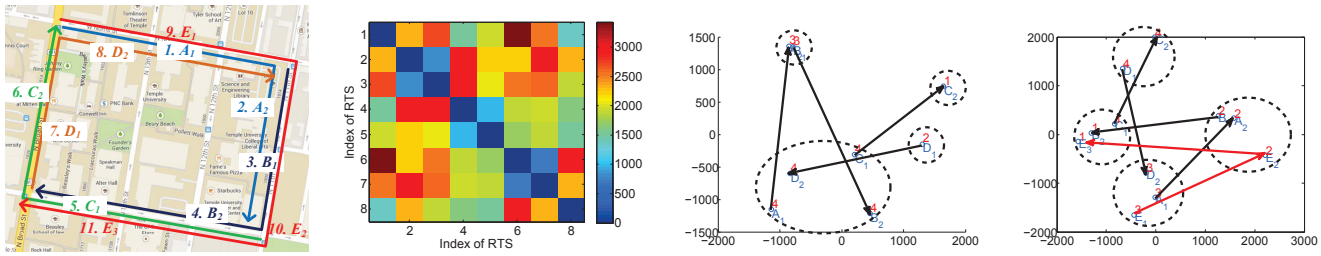
After matching the road where the querying user moves, RALS utilizes the travel times among the peaks of an RTS for estimating the speed and the position of the querying user on the matchable road. The peaks of an RTS are denoted by the sequence of RSSI as: $peaks = [pk_1, pk_2, \cdots, pk_u]$, where pk_i denotes the i^{th} peak. The server aligns and matches the peak sequence of RTS from the querying user with the peak sequence of RTS in the RTS map. For example, the two peak sequences of RTS in Figure 2(a) and 2(b) are $[-67, -64, -73, -48, -53]$, and $[-48, -79, -52, -53]$, obtained by the way of sliding window. The server aligns the two peak sequences by the algorithm of DTW. The aligned index sequences are $[1, 2, 3, 4, 5]$ and $[1, 2, 2, 3, 4]$, so the aligned two peak sequences are $[-67, -64, -73, -48, -53]$ and $[-48, -79, -79, -52, -53]$. For the duplicate peaks (such as -79 in the second peak sequence), the server reserves the aligned pair between the two sequences with the minimum distance. As a result, the two aligned peak sequences are $[-67, -64, -73, -48, -53]$ and $[-48, \backslash, -79, -52, -53]$, where $\backslash$ denotes the vacancy because of non-matchable peak.

For two aligned peak sequences on the same road, let $t_{ij}^{(q)}$ denote the travel time of the querying user between the two peaks pk_i and pk_j, and let $\bar{t}_{ij}^{(s)}$ denote the average travel time between the two peaks pk_i and pk_j from the uploaded RTS on the back-end server. The average speed between the two peaks pk_i and pk_j from the uploaded RTS on the back-end server is denoted by $\bar{v}_{ij}^{(s)}$. Because the peaks of RTS on the same road appear at the same positions, the distances among them are fixed. Thus, the ratio (denoted by $r_{q,s}$) between the speed of the querying user (denoted by $v_{ij}^{(q)}$) and the average speed in the server can be calculated by their travel times between the same pair of peaks, i.e. $r_{q,s} = v_{ij}^{(q)} / \bar{v}_{ij}^{(s)} = \bar{t}_{ij}^{(s)} / t_{ij}^{(q)}$.

If the interval time of the querying user after leaving from the peak pk_j is denoted by $\Delta t_j^{(q)}$, the server can locate the user with the interval time $\Delta t_j^{(s)}$ leaving from the peak pk_j on the RTS map, which can be calculated as: $\Delta t_j^{(s)} = \Delta t_j^{(q)} \cdot r_{q,s}$. The server can also estimate the moving distance from the peak pk_j with the average speed as: $l_j = \bar{v}_{ij}^{(s)} \cdot \Delta t_j^{(s)}$. On the contrary, for the arrival time applications as discussed in Section II-A, if the average interval time on the RTS map from the peak pk_j to the target place is denoted by $\Delta t_j^{(s)}$, the interval time of the querying user from the peak pk_j to the target place can be estimated as: $\Delta t_j^{(q)} = \Delta t_j^{(s)} / r_{q,s}$.

III. Building Map with Crowdsensing

In this section, we first introduce the approach of building the RTS map with crowdsensing. Then, we investigate the uncertain fingerprints in map construction, termed as the jigsaw puzzle problem. To speed up the map construction, we propose an approach called extra mile, which employs the regular unintentional users and a few advanced intentional users.



(a) Trajectories on road map (b) Distance matrix $\mathcal{D}$ (c) Clustering mapped in 2D graph (d) Revised clustering

Fig. 6. An example of clustering the RTS

A. Building RTS Map without Site Survey

The back-end server of RALS manages the RTS map. Due to the drawbacks of the map construction with site survey mentioned above, that of RALS is without site survey. On the back-end server, the RTS map is built by the uploaded RTS from probing users. The mobile users upload the RTS along their trajectories to the back-end server, and the uploaded RTS is divided into several segments by the intersections of roads. The server clusters multiple RTS based on their similarities, and the similar RTS are put in the same cluster, which are regarded as the same road segments. As shown in Figure 6(a), four probing users (A, B, C, and D) upload the RTS along their trajectories to the server, as an intuitive example. Each of their trajectories covers two road segments. For example, A_1 and A_2 are the RTS of two road segments along A's trajectory. When the server harvests the RTS by the way of crowdsensing, it calculates the pairwise RTS differences (or dissimilarities) δ, and stores them as a distance matrix, denoted by $\mathcal{D}$. $\mathcal{D}$ is a n-by-n matrix, where n denotes the number of the RTS. The matrix is a symmetric, square matrix with zeros along the diagonal, and each cell $\mathcal{D}(i,j)$ denotes the Euclidean distance between the i^{th} and the j^{th} uploaded RTS mentioned above. Figure 6(b) shows the distance matrix of the RTS uploaded by the four probing users. The lattice with the cold color (blue) indicates the small distance between the two RTS, and that with the warm color (red) indicates the large distance.

For visualization, the server can map these RTS into a d-dimensional space by multidimensional scaling (MDS) [22], based on the distance matrix $\mathcal{D}$. MDS is a set of related statistical techniques often used in information visualization for exploring similarities or dissimilarities in data. Seeing inter-device distances as a metric of dissimilarity, many approaches of network localization adopt MDS as a tool for calculating the locations of wireless devices [7]. An MDS algorithm starts with a matrix of item-item dissimilarities, then assigns a location to each item in d-dimensional space, where d is specified a priori. MDS returns the coordinates in a d-dimensional space for these RTS, and the RTS of each road segment is denoted by an RTS node. The resulting locations of RTS for the example are displayed in a 2D graph, as shown in Figure 6(c). Each RTS node denotes the RTS of a road segment, and each directional edge denotes the order of the two RTS in the same trajectory.

The server clusters these RTS in the d-dimensional space by applying k-means clustering [23] (a classic clustering method), which is chosen due to its computational efficiency. Here, k is set to be the number of road segments in a real map. In Figure 6(c), the integer number near each node denotes the index of the cluster which the node belongs to, by k-means clustering. The arrows denote the order of the RTS nodes along the trajectories. The RTS in the same cluster is regarded as being from the same road segment. However, we notice that A_1, B_2, C_1, and D_2 are put into the same cluster, which is not correct. This is because the dissimilarity of RTS between the two road segments is not clear enough for distinguishing by k-means clustering, due to the RSSI fluctuation problem. We term the problem of the uncertain fingerprints as the *jigsaw puzzle problem*, which need more RTS from the unintentional users, and slows down the map construction.

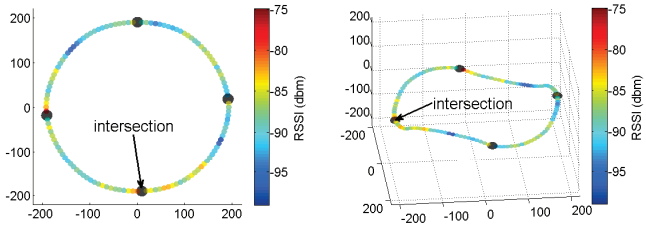
B. Jigsaw Puzzle Problem with Uncertain Fingerprints

In the map construction of RALS, each road segment can be regarded as a puzzle piece. While two RTS are adjacent in a trajectory, they are two compatible puzzle pieces, which can be merged into a bigger RTS. However, due to the RSSI fluctuation problem, the RTS of different road segments may be similar, which causes the false positive problem in the clustering of the map construction, as a problem of jigsaw puzzle. In k-means clustering, we set k as the number of road segments in this area. Thus, if the RTS from the different road segments are allocated in the same cluster (i.e. false positive), some other RTS from the same road segments can be divided into two clusters (i.e false negative).

As a result, our goal is to detect the uncertain clusters with uncertain fingerprints. We define a confidence value to measure each cluster obtained by k-means clustering. The confidence value of the i^{th} cluster is calculated as: $cv_i = e^{-\frac{\tau_i}{\alpha}}$, where τ_i denotes the average distance of all elements in the i^{th} cluster to its centroid, and α is an coefficient. The confidence value of a cluster is determined by the dissimilarity (or distance) among the RTS in the cluster. If the confidence value is smaller than a predefined threshold, we term the cluster with the low confidence value as an uncertain cluster.

After detecting the possible uncertain clusters, the system needs to distinguish whether the RTS nodes of each cluster are from the same road segment. Thus, we propose a crowdsensing-based scheme, and the basic idea is that the trajectory with non-overlapped road segments, which covers the road segments with similar RTS, can help to distinguish these RTS in the corresponding uncertain cluster.

We define the *size of an area* as the number of the road segments in this area, and define the *length of a trajectory* as the number of non-overlapped road segments covered by this



(a) 2D graph (b) 3D graph

Fig. 7. Virtual temporal map with four road segments

trajectory. Thus, in an area with a size of k, the number of the possible similar pairs is equal to $\binom{k}{2} = \frac{k(k-1)}{2}$. For a trajectory with a length of m, the number of the possible distinguishing pairs is equal to $\binom{m}{2} = \frac{m(m-1)}{2}$. We define the *certainty* as the probability of distinguishing a pair for a trajectory in the area, which can be calculated as follows:

$$Certainty(m,k) = \frac{\binom{m}{2}}{\binom{k}{2}} = \frac{m(m-1)}{k(k-1)}$$

Theorem 1: The certainty for a long trajectory with m non-overlapped road segments is larger than the sum of certainties of g short trajectories, with $m_1, m_2, \cdots, m_g$ road segments, respectively, where $m \geq \sum_{i=1}^{g} m_i$, and $m_i > 0$. i.e. $Certainty(m,k) > \sum_{i=1}^{g} Certainty(m_i,k)$.

Proof:

$$\begin{aligned} & Certainty(m,k) - \textstyle\sum_{i=1}^{g} Certainty(m_i,k) \\ = & \frac{m(m-1)-\sum_{i=1}^{g}[m_i(m_i-1)]}{k(k-1)} \geq \frac{m^2-\sum_{i=1}^{g}[m_i^2]}{k(k-1)} \\ \geq & \frac{(\sum_{i=1}^{g} m_i)^2-\sum_{i=1}^{g}[m_i^2]}{k(k-1)} = \frac{\sum_{1\leq i,j\leq g, i\neq j}[2m_i m_j]}{k(k-1)} > 0 \end{aligned}$$

Thus, $Certainty(m,k) > \sum_{i=1}^{g} Certainty(m_i,k)$. ■

Therefore, short-distance trajectories cause difficulty in distinguishing the two distant roads with similar RTS, which cause the jigsaw puzzle problem.

For the i^{th} possible similar pair of RTS, its probability of being distinguished by the trajectory X with a length of m, can be calculated as: $P_i = Certainty(m,k)$. Thus, the entropy of the trajectory X with a length of m in the area with a size of k, can be calculated as follows:

$$H(X) = -\sum_{i=1}^{\frac{k(k-1)}{2}} P_i \log_2 P_i = -\frac{m(m-1)}{2} \cdot \log_2 \frac{m(m-1)}{k(k-1)}$$

We notice that for a fixed size of the area, the shorter trajectory has larger entropy, i.e. higher uncertainty of being distinguished among the uncertain fingerprints.

C. Speeding up the Map Construction

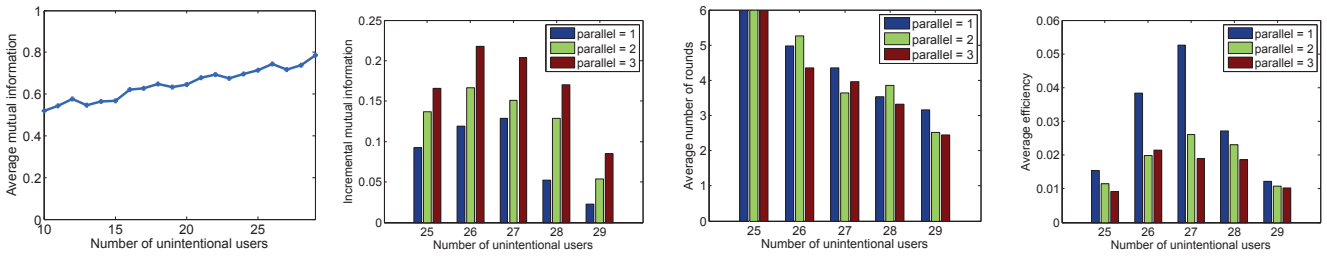
RALS employs some regular unintentional users in the map construction. The regular users sense RTS unconsciously along their normal trajectories, with the cheapest costs. As mentioned above, many studies propose that the mobile nodes with social characteristics generally visit community homes frequently, while other locations are visited less frequently [12]. As a result, the distances of the trajectories from the regular unintentional users are generally short. To speed up the map construction, RALS employs a small number of the intentional users with an additionally longer trajectory, termed as *extra mile*. The issues of the incentive schemes for mobile phone sensing [24] is out of the scope of the paper.

For an intentional user, the challenging issue is to design a long trajectory without overlapped roads. One way is to hire the mobile node moving on the pre-defined trajectory without overlapped roads, such as the passengers on a bus. Another way is to help the intentional user randomly move without overlapped roads solely by the commonly available cheap sensors in the mobile phones, which combines the following information of sensing: (1) sensing the intersections as in the aforementioned introduction; (2) sensing the landmarks at the intersections, such as cell-ID, surround sensing [18]; (3) sensing the turning direction at each intersection with the cardinal directions, by acceleration sensors and orientation sensors with arbitrary orientations of the mobile phones, which has been presented in [20]. Consequentially, the mobile phones of the intentional users record the landmarks and the turning directions of all the intersections along their trajectories, in order to avoid moving onto the overlapped road segments, which have the same entrance intersection and turning direction.

The cost of intentional users should be higher than that of unintentional users. Let $c_i(m,k)$ denote the cost of the trajectory with a length of m in an area with a size of k, which is defined as: $c_i(m,k) = \frac{a^m}{k}$, where a is the coefficient with a small value between 1.3 and 2. For the selection of the coefficient a, we define a *cost-effectiveness ratio* as the ratio between the certainty and the cost of the trajectory with a length of m in the area with a size of k, i.e. $Certainty(m,k)/c_i(m,k)$. We notice that: (1) a longer trajectory has a higher certainty, but the cost-effectiveness ratio is not monotonically increasing with the length of trajectory; (2) the small size of the area has a higher cost-effectiveness ratio. Thus, in order to narrow the area for the intentional users, their trajectories should have the same cell-IDs as the RTS in the uncertain cluster. Because the mutual coverage range of these cell-towers is limited, the size of the area is also limited.

As shown in Figure 6(a), the system additionally employs an intentional user E, who travels a trajectory with the length of 3, i.e. E_1, E_2, E_3. After k-means clustering, the RTS nodes A_1, B_2, C_1, D_2, E_1, and E_3 are put in the same cluster. However, due to E's trajectory with non-overlapped road segments, the RTS nodes E_1 and E_3 should be partitioned into two different clusters. According to the dissimilarities from E_1 and E_3 in the distance matrix, the other RTS nodes in this cluster are also partitioned into two clusters. The revised result of clustering is shown in Figure 6(d).

For each road segment, the server calculates the average travel time and the average RTS, with the aligned RTS in the corresponding cluster by DTW. We assume all the road segments are bi-directional, and both directions on the same road segment have the same average moving speed. The distance between two samples along the RTS of the same road segment is defined as the travel time between them. After clustering, the server obtains the adjacency among the clusters (i.e the road segments), with the help of the orders of RTS along the trajectories, which is denoted by the arrows in Figure 6(d). Thus, all the road segments are connected, and



(a) Quality of unintentional users (b) Incremental quality of intentional users per round (c) Number of rounds with intentional users (d) Average efficiency of intentional users

Fig. 8. Performance of map construction with crowdsensing

TABLE I. EXPERIMENT SCENARIO

Route	Description	Distance	Speed
HCW	Highway from Chengdu to Wengjiang, China	5 km	63 km/h
HCC	Highway from Chengdu to Chongzhou, China	25 km	80 km/h
SJU	Campus shuttle bus of SJU, China	3 km	22 km/h
CDC	Roads at the downtown of Chengdu city, China	5 km	28 km/h
Temple	Main campus of Temple University, USA	5 km	5 km/h

the distance between two samples from the RTS of different road segments is calculated by the shortest path based on the travel time along the RTS. Based on the pairwise distances among the samples of all the road segments, the server can obtain a virtual temporal map by MDS, as shown in Figure 7. In the map, each small node denotes a sampling point along the RTS, and its color denotes the RSSI at the sampling point. The big black nodes denote the intersections.

IV. IMPLEMENTATION AND EVALUATION

In this section, we first present our experiment environment and methodology. We evaluate the map construction with crowdsensing, the impact of the length of RTS on matching, and analyze the accuracy of localization.

A. Experimental Methodology

We implement a prototype system on the Android platform with different types of mobile phones, and collect the real data over a 3-month period in years 2012 and 2013.

1) Experimental Scenario: We did the experiments at five different places, which are listed in Table I. The routes HCW and HCC are the highways with an average speeds of 63 km/h and 80 km/h. The route SJU is the campus bus in SJU with an average speed of 22 km/h. The route CDC is on the roads of downtown Chengdu, with an average speed of 28 km/h. The route Temple is on the roads of Temple University's main campus, by walking with the average speed of 5 km/h.

2) Mobile Phone: We have tested different mobile phones are as follows: (1) China TeleCom in China: Huawei C8650 with Android 2.3, HTC S710d with Android 4.0, Samsung Galaxy SIII with Android 4.1; (2) T-Mobile in USA: Samsung Galaxy SIII with Android 4.1, Motorola MB860 with Android 2.3, Google Nexus 4 with Android 4.3.

3) Back-end Server: We implement the back-end server on the Dell DCD0, with Intel Xeon X5473 3.0 GHz Processor, and 16GB Memory. The localization service can be implemented in a computing cloud, which has been discussed in [10].

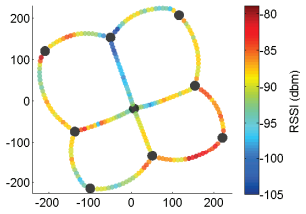
B. Map Construction

We did the experiments on the main campus of Temple University, which includes 12 one-way roads. The trajectory of each unintentional user includes two road segments, and the trajectory of each intentional user includes four road segments. We compare the sequential and parallel patterns for advanced intentional users. The system hires a single intentional user during each round in a sequential pattern, and hires multiple intentional users during each round in a parallel pattern. We limit the number of rounds to 6. To evaluate the quality of the clustering for the map construction, the metric of *mutual information* is introduced [25], [26], which is based on the Shannon information theory. Let the set $\{x_i\}$ denote the cluster assignments by RALS, and let set $\{y_i\}$ denote the correct results generated by GPS; then mutual information shows how much information is mutual for the information from $\{x_i\}$ and $\{y_i\}$. The mutual information is equal to 0 if the cluster assignment is totally different with the correct result, and it is equal to 1 if the cluster assignment is consistent with the correct result, which is the goal of map construction.

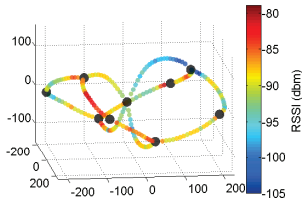
Figure 8(a) shows the mutual information (or quality) of map construction by only unintentional users. We notice that the mutual information by unintentional users increases slowly, due to the uncertain clusters. Figure 8(b) shows the incremental mutual information by adding intentional users after the map construction with different numbers of unintentional users, which is compared with the sequential and the parallel patterns during each round. We notice that large numbers of parallel intentional users can improve the quality of map construction. Figure 8(c) shows the number of rounds by adding intentional users. We notice that large numbers of intentional users during each round can reduce the number of rounds, and speed up the map construction. In addition, with more unintentional users, the number of rounds decreases. Figure 8(d) shows the average *efficiency*, defined as the average ratio between incremental quality and total number of intentional users. Although parallel intentional users can increase the incremental quality and reduce the time of map construction, the efficiency is decreased, due to the increasing number of intentional users. For visualization, we plot the 2D and 3D virtual temporal maps of the 12 road segments by MDS according to the results of map construction, as shown in Figure 9.

C. Impact of the Length of RTS on Matching

In the locating phase of RALS, the server should track the road where the querying user is currently moving, by matching



(a) 2D graph



(b) 3D graph

Fig. 9. Virtual temporal map with twelve road segments

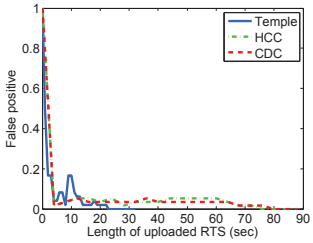


Fig. 10. Matching the roads

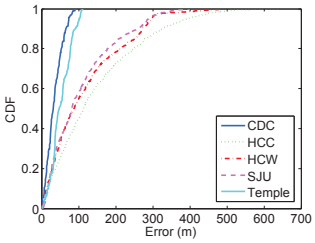
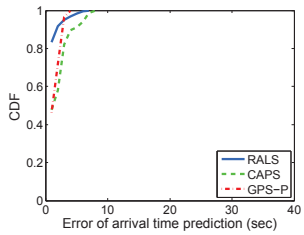
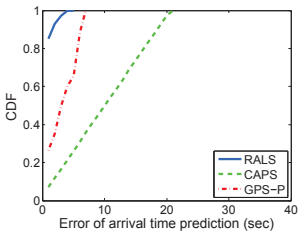


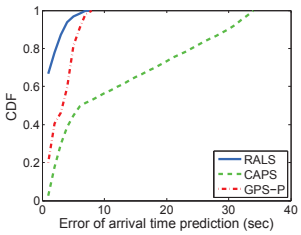
Fig. 11. Error of physical distance



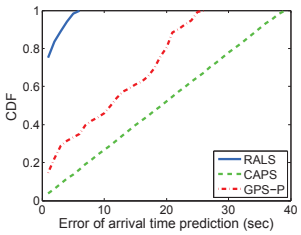
(a) HCC



(b) HCW



(c) SJU



(d) Tunnel

Fig. 12. Error of arrival time prediction

the RTS uploaded from the query user with the RTS map. We evaluate the impact of the length of the uploaded RTS on the accuracy of matching for tracking the query user. We utilize *false positive* for evaluating the performance of matching between the uploaded RTS and the RTS map. The experiment scenarios include three different types of roads, which are the highway (HCC), the urban city (CDC), and the campus (Temple). Figure 10 shows the false positive for matching over different lengths of uploaded RTS. We notice that the false positive of matching is decreasing, while the length of uploaded RTS is increasing, due to the more information for matching. The performances on different roads are different, because of the different speeds on these roads.

D. Accuracy of Localization

In the locating phase of RALS, the server aligns two matchable RTS by DTW. We evaluate the accuracy of the alignment, which can be regarded as its accuracy of localization. The outdoor localization with GPS has the highest accuracy, which can serve as the benchmark for our energy-efficient localization. We define *location error* as the average real distance of the GPS positions of the aligned pairs of points from the two matchable RTS. Figure 11 shows the location error under different scenarios. We notice that the average distance error is less than 50m in CDC and Temple, because of the short distances of the road segments and the lower moving speed. The average distance error of the campus bus is about 100m, and the average distance error of the trajectories on the highway is less than 200m, because of the different moving speeds. The results are are much less than the traditional celltower-based localization mentioned above.

Moreover, we compare the error of arrival time prediction under different moving speeds, among our proposed RALS, the cell-ID sequence-based CAPS [9], and GPS-P. Here, GPS-P denotes that the prediction of arrival time is only based on the positions information obtained by GPS, without any temporal information. We randomly select five points along each trajectory at different places as the target points, for the prediction of the arrival time, and compare it with the actual arrival time from the trajectory. Thus, we define the error of arrival time prediction as the difference between the arrival time by the localization system and actual arrival time, and the results are shown in Figure 12. We notice that the error of RALS is much less than CAPS, and the accuracy of RALS is more than two times than that of CAPS. The error of RALS is better than GPS-P, because RALS also considers the different speeds of mobile users. Particularly, in the tunnel, the performance of GPS-P is much worse than that of RALS, because the GPS device of the mobile user cannot receive the satellite signals for localization.

V. Related Work

Wireless Localization: Recently, much research has focused on the problem of locating and tracking mobile users, which includes indoor localization and outdoor localization. Bahl and Padmanabhan in [27] propose a radio-frequency based system (RADAR) for locating and tracking users inside buildings, which uses signal strength information gathered at multiple receiver locations to triangulate the user's coordinates. Constandache *et al.* in [21] design and implement Escort, a system that guides a user to the vicinity of a desired person in a public place. Shen *et al.* in [16] define WiFi-Marks, which are special pathway locations at which the trend of the received WiFi signal strength changes from increasing to decreasing, when moving along the pathway.

GPS-less Outdoor Localization: Many studies [21] have drawn attention to the tradeoff between energy and location accuracy. While GPS is fairly accurate, its continuous usage can drain a phone's battery in less than 8 hours. Thus, Paek *et al.* in [4] present RAPS, a rate-adaptive positioning system for smartphone applications. It is based on the observation that GPS is generally less accurate in urban areas. For energy-

efficient outdoor mobile localization, Paek *et al.* in [9] present CAPS, which is based on the cell-ID sequence of currently connected cell-towers along the users' trajectories. Wang *et al.* in [8] propose a continuous system location service for outdoor scenarios (Wheeloc), which relies solely on commonly available cheap sensors such as accelerometer and magnetometer.

Crowdsensing: Mobile phone sensing is a paradigm which takes advantage of the pervasive smart-phones to collect and analyze data beyond the scale of what was previously possible. Yang *et al.* in [7] investigate novel sensors integrated in modern mobile phones, and leverage user motions to construct the radio map of a floor plan, which was previously obtained only by site survey. Zhou *et al.* in [10] investigate the application of the prediction for the bus arrival time. They do not require the absolute physical location reference, and they mainly wardrive the bus routes and record the sequences of observed cell-tower IDs, which reduces the initial construction overhead.

VI. Conclusion

For an energy-efficient and accurate mobile localization, we design RALS, an RTS assisted localization system. It is divided into the training and locating phases. In the training phase, the back-end server harvests the RTS from the probing user by the way of crowdsensing, and builds the RTS map, based on the similarities among the RTS. However, the trajectories of the regular unintentional users are often limited to being short, and the map construction with short trajectories can cause the jigsaw puzzle problem. The jigsaw puzzle problem slows down the map construction, and affects its efficiency. Thus, RALS hires a small number of the advanced intentional users, who can move a longer distance for collecting the RTS, in order to speed up the map construction. In the locating phase, by matching the similar RSSI time series, RALS can track the road where the querying user currently moves; by considering the travel times among the peaks along the RTS, RALS can locate the position of the querying user on the matchable road. Our extensional experiments verify the effectiveness of our outdoor mobile localization system. In our future work, we will develop RALS for more complicated scenarios.

Acknowledgment

This work is supported by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and C-CF 1028167, and NSFC grants No. 61170256, 61103226, 61173172, 61272526, 61370204, 61472068, 61173171, and the Fundamental Research Funds for the Central Universities No. ZYGX2013J077, ZYGX2013J067, and the Applied Basic Program of Sichuan Province of China No. 2014JY0192, and the Project funded by China Postdoctoral Science Foundation (Grant No.2014M550466).

References

[1] "List of countries by number of mobile phones in use." [Online]. Available: http://en.wikipedia.org/wiki/

[2] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proc. of ACM SenSys*, 2008.

[3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proc. of ACM MobiSys*, 2008.

[4] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proc. of ACM MobiSys*, 2010.

[5] A. Thiagarajan, L. Ravindranath, K. Lacurts, S. Toledo, J. Eriksson, S. Madden, and H. Balakrishnan, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. of ACM SenSys*, 2009.

[6] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proc. of NSDI*, 2011.

[7] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proc. of ACM MobiCom*, 2012.

[8] H. Wang, Z. Wang, G. Shen, F. Li, S. Han, and F. Zhao, "Wheelloc: Enabling continuous location service on mobile phone for outdoor scenarios," in *Proc. of IEEE INFOCOM*, 2013.

[9] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-id sequence matching," in *Proc. of ACM MobiSys*, 2011.

[10] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing," in *Proc. of ACM MobiSys*, 2012.

[11] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *Proc. of ACM MobiCom*, 2012.

[12] J. Wu, M. Xiao, and L. Huang, "Homing spread: Community home-based multi-copy routing in mobile social networks," in *Proc. of IEEE INFOCOM*, 2013.

[13] W. Chang, J. Wu, and C. C. Tan, "Cooperative trajectory-based map construction," in *Proc. of IEEE TrustCom*, 2012.

[14] G. Chandrasekaran, T. Vu, A. Varshavsky, M. Gruteser, R. P. Martin, J. Yang, and Y. Chen, "Tracking vehicular speed variations by warping mobile phone signal strengths," in *Proc. of IEEE PerCom*, 2012.

[15] Y. Kim, H. Shin, and H. Cha, "Smartphone-based wi-fi pedestrian-tracking system tolerating the rss variance problem," in *Proc. of IEEE PerCom*, 2012.

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

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

[18] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *Proc. of ACM MobiCom*, 2009.

[19] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. of ACM SenSys*, 2008.

[20] C. Song, J. Wu, M. Liu, H. Gong, and B. Gou, "Resen: Sensing and evaluating the riding experience based on crowdsourcing by smart phones," in *Proc. of IEEE MSN*, 2012.

[21] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see bob?: human localization using mobile phones," in *Proc. of ACM MobiCom*, 2010.

[22] I. Borg, *Modern multidimensional scaling: Theory and applications*. Springer, 2005.

[23] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 281-297, 1967.

[24] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proc. of ACM MobiCom*, 2012.

[25] H. Zheng and J. Wu, "Spectral graph multisection through orthogonality," in *Proc. of ACM MultiClust*, 2013.

[26] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," *Physical review E*, vol. 80, no. 5, p. 056117, 2009.

[27] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proc. of IEEE INFOCOM*, 2000.