2016 IEEE 36th International Conference on Distributed Computing Systems

WiLocator: WiFi-sensing based Real-time Bus Tracking and Arrival Time Prediction in Urban Environments

Wenping Liu*†‡, Jiangchuan Liu†, Hongbo Jiang*, Bicheng Xu†, Hongzhi Lin*, Guoyin Jiang‡, and Jing Xing‡
*Huazhong University of Science and Technology, China
Email: {wenpingliu2009,hongbojiang2004, eihongzhilin2012}@gmail.com
†Simon Fraser University, Canada
Email: {jcliu, bichengx}@sfu.ca
‡Hubei University of Economics,China
Email: Jiangguoyin@hbue.edu.cn, jingxing0618@gmail.com

***Abstract*—Offering the services of real-time tracking and arrival time prediction is a common welfare for bus riders and transit agencies, especially in urban environments. On the down side, the traditional GPS-based solutions work poorly in urban areas due to urban canyons, while the location systems based on cellular signal also suffer from inherent limitations. In this paper, we present a powerful tool named Signal Voronoi Diagram (SVD) to partition the radio-frequency (RF) signal space of WiFi Access Points (APs), distributed where a bus travels, into Signal Cells, and then into fine-grained Signal Tiles, tackling the problem of noisy received signal strength (RSS) readings and possible AP dynamics. On top of SVD, we present a novel framework so-called WiLocator, to track and predict the arrival time of an urban bus based on the surrounding WiFi information collected by the commodity off-the-shelf (COTS) smartphones of bus riders, the mobility constraint of a bus and the temporal consistency of travel time of buses on the overlapped road segments. We also show the WiLocator's power of generating an accurate and real-time traffic map with the predicted travel time on each road segment. We implement the prototype of WiLocator and conduct the in-situ experiment to demonstrate its accuracy.**

I. Introduction

Urban is where most of the important transportations (such as airports and rail yards, etc.) terminate, making the urban transit crucial to efficiently support the mobility of people and freight. For instance, in 2014 alone, American people (most of them are young) took about 10.8 billion trips on public transportation, and, around 70% of population of Mexico takes public transport. As such, the habitual mobility of people gives rise to the recent observation that the urban productivity highly depends on the pubic transport system. In this paper we focus on one type of the pubic transport system, the bus system, as it covers the majority of the urban road network, compared with other transport modes such as skytrain or subway. For instance, in London, about 75% road segments are covered by bus systems, and 79% in Singapore, and around 85% of residents have a bus stop within 400 meters of their home in a metropolitan of North America. It is noted that the bus system suffers more from traffic jam, especially in rush hours. A long waiting time for a bus clearly discourages people to take bus for commute, shopping, etc. The information, if available, of where the bus is and when it will get the intended stop, no doubt can cut down the waiting time, and thus increase the efficiency of bus riders [19], which in turn appeals more and more people to take bus transportation, and thus benefits the transit agencies from offering the services of real-time tracking and arrival time prediction. In summary, offering the services of real-time tracking and arrival time prediction is a common welfare for bus riders and transit agencies.

To provide these services, many transit agencies (e.g., TTC[1] and CTA[2], etc.), or third-party companies (e.g., NextBus[3]), have commonly equipped each bus with a GPS-enabled in-vehicle device (so-called Automatic Vehicle Locator, or AVL unit). Unfortunately, the adoption of GPS based tracking and arrival time prediction in real world has been stymied by the GPS-based device's well-known limitations on energy consumption, requirement of sky visibility, initial and incurring costs which prohibit the applicability for small-scale transit agencies, and so on. An alternative approach relies on cellular infrastructure [15], [20], [21], [27]–[29]. However, the long capturing time for a stable Cell-ID sequence, the overlapped road segments of different routes, low density of cell tower, etc., make the cellular-based approaches a poor fit for real-time bus tracking.

WiFi Access Points (APs), on the other hand, are now distributed densely along the road segments of the urban bus routes. Fig. 1 gives a glance of WiFi hotspot distribution in a big city of north America. It's well-known that the noisy RSS readings entail substantial challenges for positioning. For instance, the existing WiFi-based localization systems either rely on fingerprinting the receive signal strength (RSS) [2], [3], [5], [9], [10], [16], [17], [24]–[26], or the signal propagation model [6]. However, the calibration for fingerprint database requires intensive labors from experts in-house, and also suffers from the dynamics of WiFi APs due to reconfiguration or replacement, etc., while the radio-frequence (RF) propagation model-based schemes hurt seriously the positioning accuracy.

Observing that the average RSS rank from an AP sensed by multiple devices remains relatively stable, in this paper, we present a powerful tool, named *Signal Voronoi Diagram*

[1]http://www.ttc.ca.
[2]http://www.chicagobus.org/news/bus-tracker-expands-systemwide.
[3]http://nextbus.cubic.com.

1063-6927/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDCS.2016.31

(SVD), to partition the signal space into coarse-grained Signal Cells (SCs), within each of which the points receive the strongest signal from the same AP (referred to as *site* or *generator*), and then into fine-grained Signal Tiles (STs). Within each ST, the rank of RSS readings remains constant. On top of SVD, we exploit the mobility constraint of the bus and temporal consistency of travel time of buses on overlapped road segments, and design a novel and energy-efficient framework, so-called WiLocator, to real-time track and predict the arrival time of the bus in urban environments.

WiLocator consists of three components: (1) WiFi-enabled commodity off-the-shelf (COTS) smartphones, carried by the driver and bus riders for crowd sensing. The smartphone periodically scans the surrounding WiFi information, and reports it to the server; (2) a back-end server (we shift the computation burden to the server, including SVD construction and real-time bus tracking, arrival time prediction, traffic map generation and anomalies detection); (3) a user interface for trip plan, such that the real-time bus track and schedule, and the traffic map, can be readily available for intended bus riders.

The contributions are summarized as follows:

- We propose a powerful tool, the Signal Voronoi Diagram generated by WiFi APs, to divide a signal space into coarse-grained Signal Cells and fine-grained Signal Tiles, tackling the unstable WiFi signals;
- We exploit the SVD, the mobility constraint of a bus, and the temporal consistency of travel time of the buses on the overlapped road segments, and design WiLocator, a WiFi-sensing based framework for real-time bus tracking and arrival time prediction;
- We present a novel scheme to report anomalies (if any), and generate the real-time traffic map by analyzing the statistical behaviors of the travel time on each road segment, instead of the vehicle velocity subjecting to different speed limits on different road segments;
- We implement a prototype of WiLocator and conduct the in-situ experiment to show the efficiency.

The rest of the paper is organized as follows. We present the background and motivations in Section II, and in Section III, we present the principle of the signal Voronoi diagram based bus positioning. Section IV is devoted to the applications for arrival time prediction and real-time traffic map generation. We present the prototype implementation and experimental results in Section V, and in Section VI we briefly introduce the related work. Finally, we conclude the paper and present the future work in Section VII.

II. Background and Motivations

As mentioned earlier, several transit agencies and third-party companies offer the services of real-time bus tracking and arrival time prediction by installing GPS-enabled AVL devices. However, equipping each bus with an AVL devices potentially incurs overwhelming initial and recurring costs [19], which renders this solution prohibitive for practice, especially for many small transit agencies operating only a few buses. As such, Biagioni *et al.* [4] presented EasyTracker, an automatic

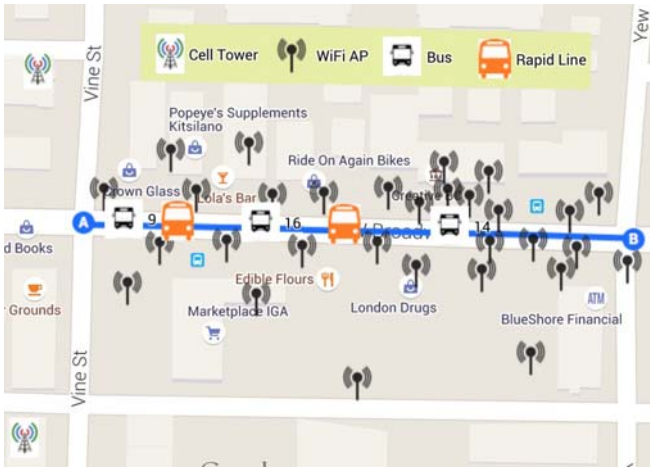


Fig. 1: The motivated example on a road segment of a main street in Metro-Vancouver, Canada. The geo-tagged WiFi APs (with their latitudes and longitudes marked in Google Maps) are densely distributed along the road where multiple bus routes share a few overlapped road segments, and there are only a small number of cell towers, each of which covers multiple road segments.

bus tracking and arrival time prediction system on a COTS smartphone, carried by the driver or installed in the bus. Unfortunately, GPS is power-hungry, and the GPS-based tracking systems including EasyTracker work poorly in urban environments due to the city geometry (or so-called urban canyons): the high buildings, or tunnels, will block the line-of-sight (LOS) paths to the satellites. The existing energy-accuracy tradeoff triggers the development of lightweight positioning systems, which either reduces the accuracy requirements down to an acceptable limit, or turns on and off GPS based on some clues, or both [7], [13]–[15], [27]–[29]. For instance, in [15], [27]–[29], the location is computed based on Cell-ID sequence matching. Though energy-friendly, they can not efficiently handle the case of overlapped road segments, and suffer from the low density of cell tower and long capturing time for a stable Cell-ID sequence for matching.

Nowadays, geo-tagged WiFi APs, with their latitude and longitude marked in Google Maps, are widely visioned in modern metropolitans: hotels, restaurants, and so on. As WiFi network is viable and has already been exploited for data offloading by mobile operators, we envision that there will be an increasing trend of WiFi APs deployment covering densely the majority of road networks in urban areas. Compared with other approaches, WiFi-based bus positioning is attractive since, 1) there are already enough WiFi APs with limited coverage due to the limited transmitted power, in urban areas, which entails an accurate result based on RSS readings from surrounding WiFi APs; 2) it only take several seconds to retrieve the necessary WiFi information, such as SSID, BSSID and RSS, etc., around the road segments; 3) it does not suffer from city geometry, and 4) there is no installation or

maintaining fee for transit agencies. Despite these advantages, the unstable WiFi signal, AP dynamics, and the complicated outdoor environments along the road segments, where existing WiFi-based location systems relying on calibration by experts in-house or radio-frequency (RF) propagation model are inefficient, entail substantial challenges for bus tracking. In this paper we propose the powerful tool, named Signal Voronoi Diagram (SVD), to tackle these challenges.

In addition to the densely distributed WiFi APs in urban environments, another key observation is that, different bus routes, e.g., route 9, 14, 16 and a Rapid Line in Fig. 1 may share a few overlapped road segments connecting adjacent intersections/terminals. The experimental evidence shows that the travel pattern (e.g., normal, faster or slower as compared with the historical travel time) of different bus routes on the same road segment exhibits high temporal correlation: if a bus A has just travelled by a road segment at a normal travel pattern, then the travel time of next bus B, despite its route, on this road segment will also be normal with high probability, even though their regular speeds on this segment may differ. Clearly, the travel time of previous buses on the shared road segments is most timely and thus capable for offering an accurate travel time estimation of the upcoming buses. As such, by leveraging the travel patterns of buses having just passed by the intended road segment, we can accurately estimate the travel time of the upcoming buses on this road segment, and its arrival time at subsequent stops.

Motivated by these facts, in this paper, we aim to track and predict the arrival time of a bus in urban areas by participatory crowd-sensing based on WiFi-enabled smartphones, and design a cost-effective, user-friendly and reliable system, named *WiLocator*, requiring the minimum interference of participants, i.e., zero effort from the bus riders.

III. Signal Voronoi Diagram Based Bus Positioning

A. Signal Voronoi Diagram

Voronoi diagram (VD), also known as Voronoi tessellation, is the partition of a plane into regions, called *Voronoi cells*, based on the distance to the points (referred to as seeds, sites, or generators) of a finite set [1]. Formally, given a point set $S = \{p_1, p_2, \cdots, p_n\}$, the dominance of p_i over $p_j (i \neq j)$ is defined as the sub-plane not farther to p_i than to p_j. Formally,

$$dom(p_i, p_j) = \{x \in R^2 | d(x, p_i) \leq d(x, p_j)\} \quad (1)$$

where $d(x, y)$ denotes the Euclidean distance between x and y. The Voronoi cell generated by p_i, denoted by $c(p_i)$, is thus defined as the sub-plane in the dominance of p_i over all other sites. That is,

$$c(p_i) = \cap_{j=1,2,\cdots,n, j\neq i} dom(p_i, p_j) \quad (2)$$

Ties are allowed here and the intersection of two neighboring cells $c(p_i), c(p_j)(i \neq j)$, where the points are equidistant to p_i and p_j, form a *Voronoi edge* $e(p_i, p_j)$, and two Voronoi edges meet at the *junction point*. The collection of these Voronoi cells generated by the sites in S, forms a partition, i.e., the Voronoi diagram, of the plane, denoted by $VD(S)$.

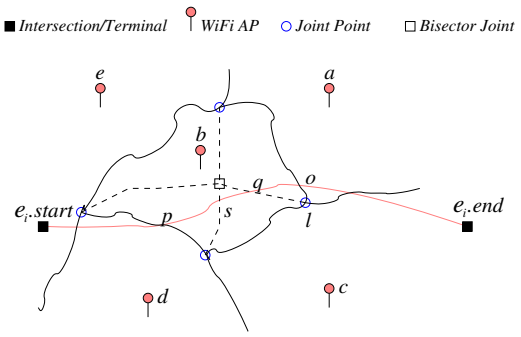


Fig. 2: SVD and its application for bus positioning. The WiFi APs, namely, a, b, c, d and e around the road segment e_i generate the SVD where the solid lines represent SVEs; Signal Cell $SC(b)$ is partitioned into four STs. Points s, p, q and o are the intersection of road segment e_i with the $SVE(b, d)$, the tile boundary of $ST(b, d) \cap ST(b, c)$, tile boundary of $ST(b, c) \cap ST(b, a)$ and $SVE(b, a)$, respectively.

Definition 1. *Given a WiFi AP set $P = \{p_1, p_2, \cdots, p_n\}$ in a space D, the Signal Voronoi Diagram of D, denoted by $SVD(D)$, is a partition of D into a set of Signal Cells based on $RSS(x, p_i)$, the signal strength at point x, where each cell $SC(p_i)(i = 1, 2, \cdots, n)$ is the dominance of p_i over other APs such that $SC(p_i) = \{x \in D | RSS(x, p_i) \geq RSS(x, p_j), j = 1, 2, \cdots, n, j \neq i\}$. The Signal Voronoi Edge (SVE) between p_i and p_j, denoted by $SVE(p_i, p_j)$, is a point set such that $SVE(p_i, p_j) = \{x \in D | RSS(x, p_i) = RSS(x, p_j)\}$. We call the point where two or more SVEs meet as a joint point.*

In what follows, we will use interchangeably the term of AP and site. It can be seen that the SVD differs from the VD in that the metric here is the RSS of the point from the APs, instead of the Euclidean distance as in [12], [16], [18]. As in practice, many factors, such as transmit power, the environment, etc., affect the RF signal propagation, the SVE is not necessarily a straight-line, making the VD inefficient for this case; only in the ideal case where all of these parameters are equal for all APs will the SVD be the same as the VD. Therefore, the conventional Voronoi Diagram is just a special case of SVD. Fig. 2 gives some intuition for SVD.

Note that within each SC, the difference of RSS between other sites, instead of the site of this SC, is ignored. For instance, for any $x \in SC(p_i)$, the relationship, or the rank order, between $RSS(x, p_j)$ and $RSS(x, p_k)(i \neq j \neq k)$ is unclear. To that end, we can further partition each SC into fine-grained regions, which we call as the *Signal Tiles* (STs).

Definition 2. *Given the Signal Cell $SC(p_i)$ having n_i neighboring SCs with sites $P_i = \{p_{n_1}, p_{n_2}, \cdots, p_{n_i}\}(n_j \neq i, j = 1, 2, \cdots, i) \subseteq P$, a Signal Tile $ST(p_i, p_{n_j})(j = 1, 2, \cdots, i)$ of $SC(p_i)$ is defined as the dominance of p_{n_j} in $SC(p_i)$ over other sites in P_i. That is, $ST(p_i, p_{n_j}) = \{x \in$*

$SC(p_i)|RSS(x,p_{n_j}) \geq RSS(x,p_{n_k}), k = 1,2,\cdots,i,k \neq j\}$, *or* $ST(p_i,p_{n_j}) = \{x \in D|RSS(x,p_i) \geq RSS(x,p_{n_j}) \geq RSS(x,p_{n_k}), k = 1,2,\cdots,i,k \neq j\}$ *where at least one inequality holds as the Signal Tile is not empty. We call the intersection of two neighboring STs as Tile Boundary, and the intersection of two or more tile boundaries as a Bisector Joint.*

Clearly, the collection of the STs in each SC then form a finer partition of D, i.e., the second-order SVD, and thus each ST is also called as a second-order Signal Cell. We can derive the higher-order SVD by similarly conducting the partition process for each Signal Tile $SC(p_i)$ until it's decomposed into $n_i!$ finest Signal Tiles $ST(p_i,p_{n'_1},p_{n'_2},\cdots,p_{n'_i})$ which are no longer dividable based on the available RSS vector, where $(n'_1,n'_2,\cdots,n'_i)$ is a permutation of $(n_1,n_2,\cdots,n_i)$.

Proposition 1. *For any* $x \in ST(p_i,p_{n'_1},p_{n'_2},\cdots,p_{n'_i})$*, the following inequality holds:* $RSS(x,p_i) \geq RSS(x,p_{n'_1}) \geq RSS(x,p_{n'_2}) \geq \cdots \geq RSS(x,p_{n'_i})$*. That is, the RSS values are ordered within each ST.*

It's well-known that the RSS readings are very unstable: even in a static point it can vary up to more than 10 db, making the RSS readings less important. However, the ranks of the RSS from different APs are relatively stable, and according to Propositon 1, we can easily infer the location range of a mobile device, i.e., the ST it's in, based on its collected RSS vector from the surrounding WiFi APs, and estimate its location as the centroid of the ST. Note that here no calibration or RF propagation model is required. Clearly, the accuracy of rank-based positioning depends on the size of ST that the device resides in, which is subject to the WiFi AP density, and the order of SVD. As such, we have the following propositions:

Proposition 2. *A higher-order SVD based positioning scheme will provide a more accurate result.*

Proposition 3. *The SVD constructed based on more APs will ofter a higher positioning accuracy.*

With these features, next we will apply the constructed SVD for WiFi-sensing based real-time bus positioning.

B. Signal Voronoi Diagram based Bus Positioning

As mentioned before, there are densely distributed geo-tagged WiFi APs along the road segments of bus routes, which are sufficient to construct the SVD with small-sized STs for an accurate position. However, generally the SVD alone can not pinpoint the bus position. For instance, in Fig. 2, if the AP sequence according to the RSS rank list is (b,a,d), then we infer that the current bus position falls within the Signal Cell $SC(b)$, or more precisely, the Signal Tile $ST(b,a)$. However, the ST's centroid might be far from the ground truth which should be on the road segment. Note that a bus will follow a regular route, which can be readily downloaded from the website of the transit agency. With this mobility constraint, we can narrow down the bus position estimation to the road segment in the road network, which is defined as follows.

Definition 3. ***Road Network.*** *A road network is a directed graph* $G(V,E)$ *where* V *is the set of vertices corresponding to the intersections and road terminals, and the edge set* E *denotes the collection of directed road segments between two adjacent vertices* $v_i.start$ *and* $v_i.end, 1 \leq i \leq |V|$.

Definition 4. ***Bus Route.*** *A bus route* R *is a sequence of connected and directed road segments* $e_1 \rightarrow e_2 \cdots \rightarrow e_n$ *where the start stop* s_1 *and final stop* s_n *lie on* e_1 *and* e_n*, respectively, and* $e_i.end = e_{i+1}.start, 1 \leq i < n$.

Please see Fig. 3 for some intuition. To derive the bus position, we define a Tile Mapping[4] as follows:

Definition 5. *Let* $ST(p_i,p_{n_j})$ *be the Signal Tile that the bus currently resides in, and* e_{ij} *be a sub-segment, in* $ST(p_i,p_{n_j})$*, of road segment* e_i*. We define the Tile Mapping* $F : ST(p_i,p_{n_j}) \rightarrow e_{ij}$ *as* $F(ST(p_i,p_{n_j})) = p_{ij}$ *where* p_{ij} *is the nearest point of the centroid of* $ST(p_i,p_{n_j})$ *to* e_{ij}.

As such, based on the second-order SVD that divides a road segment into short sub-segments, we first determine which ST the bus is in, and then map this ST to the road sub-segment inside the ST to infer the bus position. For instance, in Fig 2, if the RSS rank list is (b,a,d), then the bus location is inferred to be a point on the sub-segment between o and q, and between p and s for the rank list (b,d,c). When there are more APs, the positioning result will also be more accurate, as each SC can be decomposed into more STs, and thus the road segment within each ST can be divided into shorter sub-segments.

On the downside, due to the noisy RSS readings, the bus might be mistakenly estimated to reside in a ST, e.g., $ST(b,e)$ in Fig 2, which has no intersection with the road segment. In this case, we simply map this ST to the nearest point on the road sub-segment that intersects with the neighboring ST with the longest tile boundary, and then the bus position can be inferred accordingly. For instance, the tile boundary between $ST(b,d)$ and $ST(b,e)$ is the longest among other tile boundary of $ST(b,e)$, and we thus map $ST(b,e)$ to the road sub-segment between p and s.

Note that there can be equal ranks of RSS values from different APs, and in this case the position estimation is much easier. In Fig. 2, if rank of a (or d) equals to that of b, then o (or p) will be the estimated position. And if RSS ranks from a,b,c are equal, theoretically the junction point l, where $SVE(a,b)$ and $SVE(a,c)$ meet, is the best estimation. However, the bus must travel on the road segment, therefore we project the junction point l to the road segment e_i, and regard the projected point as the estimated bus position.

Occasionally, there can be dynamics of WiFi APs, e.g., due to their being out of function, and replacement, etc., but our SVD-based positioning algorithm does not suffer from such dynamics. For instance, in Fig 2, suppose that the AP b is out of function. The surrounding WiFi APs are now becoming a,c,d and e, and the SVD changes accordingly where the Signal Voronoi Edges $SVE(b,a), SVE(b,c), SVE(b,d)$

[4]In practice, we find that a second-order SVD is enough for a high accuracy.

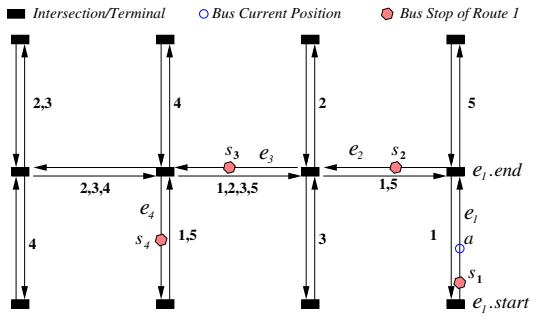


Fig. 3: Road network and the bus routes (indicated by the number).

and $SVE(b, e)$ do no longer exist. Other SVEs, i.e., $SVE(a, c), SVE(a, e), SVE(c, d)$ and $SVE(c, e)$, and the dashed tile boundary of STs in $SC(b)$, together form the new SVEs with the bisector joint (indicated by the empty rectangle) as the junction point of the new SVD. We can further construct the second-order SVD, and in a similar way described above we can infer the bus position based on the ranks of RSS from the available APs. One can imagine that the newly estimated position will not be far away from the true location.

With the estimated bus position and a time stamp indicating the time of WiFi-sensing, we derive a trajectory of the bus.

Definition 6. ***Trajectory.*** *A bus trajectory is a sequence of tuples* $< lat, long, t >$ *representing the latitude, longitude and the time stamp, respectively.*

IV. Bus Arrival Time Prediction and Real-time Traffic Map Generation

In this section, we detail the applications of SVD-based bus positioning for arrival time prediction and traffic map generation. Note that the buses with the same consecutive stops generally share overlapped road segments, while the buses sharing the same overlapped road segments do not necessarily have the same stops. As such, our scheme is built on road segments, with an advantage of leveraging more lately travel time of buses with the same/different routes to predict the arrival time, over other solutions [28], [29] that only use the data of the same route. To that end, we first fetch the historical and lately travel time on each overlapped road segment of these bus routes, and then summarize these travel times to estimate the arrival time of a bus at the subsequent stops.

Let us take Fig. 3 as an example where the current bus location of route 1 is marked by the empty circle, and the solid diamonds represent the subsequent stops of the bus route 1. After locating the bus, we find out whether there are buses sharing the overlapped road segments between the current location and the intended stop(s), and estimate the travel time from current position to the end of current road segment, and to subsequent road segments, based on the historical and lately data of these routes (if there are other routes sharing road segments with route 1). On current road segment e_1, where only buses of route 1 travel, the historical and lately (if any) data of route 1 is used since there is no bus of other routes traveling on e_1. Later, we find there are two routes, namely, route 1 and 5 share segment e_2; if buses of route 1 and 5 have just passed by e_2, we can use the historical data of route 1 and 5, and their up-to-data travel time to estimate how long the next bus of route 1 will take to travel along this road segment e_2. Next we only focus our concentration on estimating the travel time of a bus route on a single road segment.

Specifically, suppose that there are $K(> 1)$ bus routes $R_1, R_2, \cdots, R_K$ sharing the same road segment e_i. To predict the travel time of an upcoming bus, say of route R_1, on e_i, we suppose that most recently there are J buses of $K\prime(\leq K)$ routes passing by e_i. We denote by $T_h(i, j), T_r(i, j)$ the historical and recent travel time of bus $j \in (1, J)$, respectively. Note that many factors affect the travel time: weather condition, driving style, the number of boarding and alighting passengers, whether there is a stop on the road segment, and unexpected accidents, to name a few. Incorporating all these factors into a model would result in a time-expensive computation of the model. Instead, we classify them into two categories: bus route-dependent and environment-related factors. The former one is related to each bus route, while the latter is shared by all routes on the same road segment, which is often uncontrollable by any bus route and thus we model it as a random variable ϵ_i following a Gauss distribution $N(\mu_i, \sigma^2)$. Thus, the travel time of route j on e_i can be formulated as

$$T_r(i, j) = \mu_{i,j} + \epsilon_i \tag{3}$$

where $\mu_{i,j}$ is the mean of the travel time of route R_j on road segment e_i, which can be unbiasedly estimated as $T_h(i, j)$, i.e., $E(T_h(i, j)) = \mu_{i,j}$ where $E()$ denotes the expectation, for a large number of historical data. Thus, Equation 3 can be rewritten as

$$T_r(i, j) = T_h(i, j) + \hat{\epsilon}_i \tag{4}$$

where the residual $\hat{\epsilon}_i$ is an unbiased estimator of ϵ_i, namely, $E(\hat{\epsilon}_i) = \mu_i$. Then, the travel time of route R_1 on road segment e_i in the near future can be estimated as

$$T_p(i, 1) = T_h(i, 1) + \frac{\sum_{j=1,\cdots,J}\{T_r(i, j) - T_h(i, j)\}}{J} \tag{5}$$

where the second item in the right side is an estimation of $\hat{\epsilon}_i$.

Typically, there are two significant rush hours in a weekday, one in morning and one in afternoon, incurring a large variation σ^2 caused by the environment-related factors. Hence, we can divide a weekday into serval time slots, for instance, morning non-rush hour, morning rush hour, afternoon non-rush hour, afternoon rush hour, and otherwise, and estimate the average historical travel time of each bus route. Then, within each time slot, we build a prediction model in a similar way.

It is noted that the rush hour may appear at different time for different node segments. To that end, we need to figure out for each node segment when the rush hour is based on the historical data by computing the so-called *seasonal index*, a metric to determine whether there is a seasonal cycle (or, periodicity) of an economic phenomena in statistics.

Assume that each day is divided into L time slots, e.g., each hour is a time slot, and we have an M-day observation of bus travel time. During each day m, for each bus route $R_j(1 \le j \le K)$ on road segment $e_i(1 \le i \le n)$, at time slot $l(1 \le l \le L)$, the travel time is denoted by $T(i,j,m,l)$. Let $\bar{T}(i,\cdot,\cdot,l)$ denote the average travel time of all routes on road segment e_i from time slot 1 to L, and $\bar{T}(i,\cdot,\cdot,\cdot)$ be the whole average travel time with respect to the day and time slot. That is, $\bar{T}(i,\cdot,\cdot,l) = \frac{\sum_{m=1,\cdots,M,j=1,2,\cdots,K} T(i,j,m,l)}{MK}$, $\bar{T}(i,\cdot,\cdot,\cdot) = \frac{\sum_{,j=1,2,\cdots,K,m=1,\cdots,M,l=1,\cdots,L} T(i,j,m,l)}{MKL}$. The seasonal index of time slot l, denoted by $SI(i,l)$ is thus defined as

$$SI(i,l) = \frac{\bar{T}(i,\cdot,\cdot,l)}{\bar{T}(i,\cdot,\cdot,\cdot)} \tag{6}$$

Clearly, we have

$$\sum_{l=1,\cdots,L} SI(i,l) = L, SI(i,l) > 0 \tag{7}$$

If $SI(i,l) = 1$ for any $l \in [1,m]$, there is no periodicity of travel time; if $SI(i,l) \gg 1$ (e.g., $SI(i,l) \ge 1.6$ in our experiments), showing that the travel time is much longer compared with the average level, then the time slot l might be a rush hour. We can group consecutive time slots with similar seasonal index into a bigger slot such that each day can be divided into less slots, to increase the sample size for predicting the arrival time. Then, for any time t of time slot l, we can rewrite Equation 5 as:

$$T_p(i,j,t) = T_h(i,j,l) + \frac{\sum_{k=1,\cdots,K}\{T_r(i,k,l) - T_h(i,k,l)\}}{K} \tag{8}$$

Assume that at a time t within a time slot l, the bus of route j is currently on position a of segment e_i, the arrival time at stop s_n on segment $e_n(n > i+1)$ is estimated as

$$T_e(s_j) = \frac{T_p(i,j,t)d_r(p,e_i.end)}{d_r(e_i.start,e_i.end)} + \sum_{k=i+1}^{n-1} T_p(i,j,t) + \frac{T_p(n,j,t)d_r(s_n,e_n.end)}{d_r(e_n.start,e_n.end)} \tag{9}$$

where $d_r(x,y)$ denotes the road length between x and y. For $n = i+1$, the middle term of right side in Equ. 9 is ignored. When the intended road segment or stop is far away from current position such that the arrival time might fall in another time slot, the computation will be separated slot-by-slot.

V. Prototype Implementation and Experiment Results

A. Prototype Implementation

WiLocator consists of three main blocks, namely WiFi-based bus positioning, arrival time prediction, and traffic map generation, corresponding to three components for data sensing, processing and applications, as shown in Fig. 4.

- **SVD-based Bus Positioning.** We assume that the bus route can be easily identified, e.g., based on voice recognition of the announcement by riders or text input by the driver. The smartphones of the driver and bus riders then periodically scan available WiFi, and report the WiFi information to the back-end server. The server then constructs the Signal Voronoi Diagram (SVD) according to the average rank of RSS values from each of surrounding WiFi APs; by leveraging the mobility constraint of the bus, the server yields an estimated position.
- **Arrival Time Prediction.** For each subsequent road segment of a bus on its route, we leverage the historical and lately travel time of the buses of the same/different routes traveling on this segment to estimate the travel time of next bus on this road segment, and then the arrival time of the bus at the subsequent stops can be easily computed based on the estimated travel time on each road segment where a simple interpolation technique might be needed.
- **Traffic Map Generation.** Since different road segments may pose different speed limits, instead of using the vehicle velocity, we analyze the statistical behavior of the travel time on each road segment to generate a real-time traffic map and report the traffic anomalies if any.

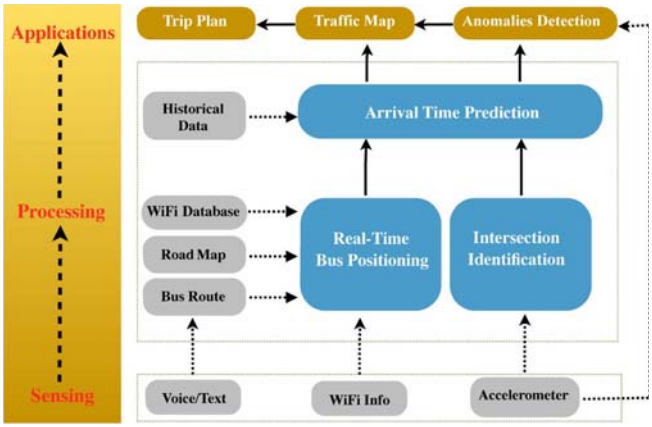
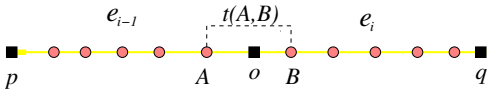


Fig. 4: The implementation diagram of WiLocator.

Note that in WiLocator, the computation burden is shifted to the server, and no interference from the driver/riders is required. Next we will detail the prototype implementation of WiLocator, followed by the experimental evaluation.

1) Bus Route Identification: The first step of WiLocator is to identify the bus route. The Cell-ID matching based scheme [28] can not correctly classify the bus route when the first few road segments overlapped with other routes, quite common in urban areas, and thus the arrival time of the stops on these road segments is hardly available. In fact, nowadays, when the bus starts, it usually announces the bus route, including the route and the destination it bounds for. Thus we can easily extract the route information and where it bounds for based on the voice recorded by riders' smartphones. On the downside, since we assume that each driver carries a smartphone installed WiLocator, the bus route can also be supposed to be known; the bus riders, close to the driver by proximity sensor, have approximately the same trajectory,



■ *Intersection/Terminal* ● *Bus Historical Position*

Fig. 5: Identification of intersections.

therefore we can easily determine which bus the riders are on.

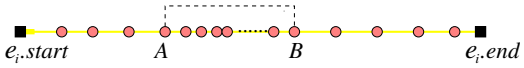
2) Real-Time Bus Tracking: The core of WiLocator is to locate the bus instantly as follows. When the system starts, the smartphone periodically scans the surrounding WiFi information, including SSID, BSSID, and RSS, and upload them, together with a time stamp, to the back-end server. After receiving the report, the back-end server ranks the RSS values and then construct the SVD to determine which SC the bus is in, followed by partitioning the SC into STs. With the route information and the road map downloaded from the transit agency and Google maps, respectively, WiLocator maps the ST which the bus resides in to the road sub-segment of the bus route, and therefore infers the location accordingly.

3) Arrival Time Prediction: WiLocator has two phases, namely offline training and online prediction for this task.

Offline training. For each road segment, the server computes the seasonal index based on the historical travel time, and determine whether there is a periodicity. If so, the server will divide the day into time-slots, within each of which the travel time of a bus route on this road segment is assumed to follow the same probability distribution.

Online Prediction. Based on the estimated position and the time stamp, the server estimates its arrival time at subsequent stops. First, it finds how many buses have travelled the subsequent road segments and computes the travel time of each bus. To that end, we need to compute when a previous bus arrived at the start point of the road segment and when it left, since there might be a gap between the WiFi scanning time and the arrival time at an intersection. Generally, there are two cases: 1) the bus stopped at the end point of the last road segment, e.g., waiting for the traffic light turning green. For this case, the arrival time at the next road segment can be easily approximated[5]; and 2) the bus went straightly from road segment e_{i-1} to another road segment e_i when the blue traffic light is on. In this case, we assume that the bus travelled smoothly, i.e., at a steady speed, and use two locations, say A on e_{i-1} and B on e_i in Fig. 5, to approximate the arrival time at the intersection, i.e., $e_{i-1}.end$ of segment e_{i-1}. Specifically, let the travel time (i.e., the scanning period) between A, B be $t(A, B)$, and the travel time from A to $e_{i-1}.end$ can then be approximated as $\frac{t(A,B)d(A,e_{i-1}.end)}{d_r(A,B)}$. As a result, the arrival time of each last bus at $e_{i-1}.end$ or $e_i.start$ can be easily estimated. With these lately data on each road segment, the arrival time of the next bus at subsequent stops can be estimated according to Equation 9.

[5]We can also use the built-in accelerometer sensor to trigger a WiFi scanning and upload the report to the server when the bus stops.



■ *Intersection/Terminal* ● *Bus Historical Position*

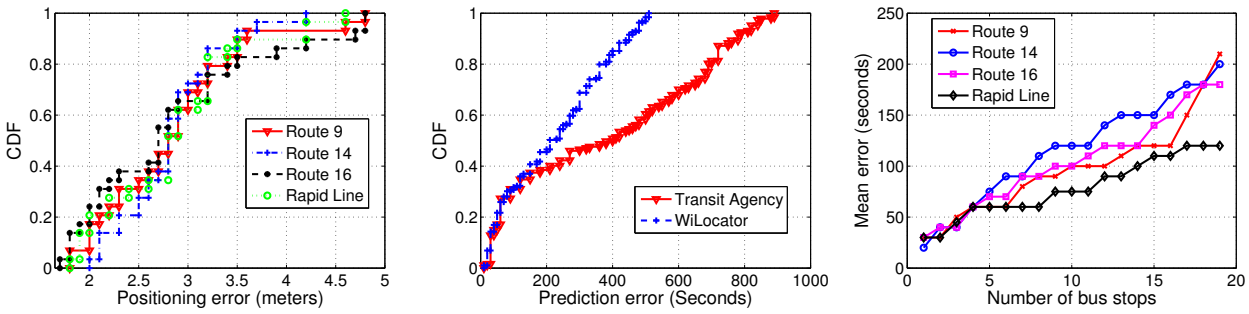
Fig. 6: An illustrative example of anomalies detection.

Fig. 7: The four bus routes (i.e., the Rapid line, 9, 14 and 16) in Metro-Vancouver, Canada. The terminals of different routes are marked by different shapes.

4) Real-time Traffic Map Generation: The traditional approach to generate a traffic map is based on the velocity of vehicle passing each road segment. However, this method is somehow problematic since each bus route usually has different regular speed when traveling the same road segment. For example, a bus of a rapid transit line (e.g., the Rapid Line in Fig. 1) usually runs faster than ordinary buses (e.g., Route 9,14, and 16 in Fig. 1); the results based on the data of rapid transit line or an ordinary bus might be misleading. Besides, different road segment, e.g., near school zone or not, may pose different speed limits. As such, WiLocator generates the real-time traffic map based on the statistics of travel time of a bus on each road segment, instead of the velocity.

Specifically, for each road segment, we determine the traffic condition based on the current and historical travel time. If the current travel time is, say, c_1 times the standard deviation (STD), larger than the mean travel time, we mark this road segment as *very slow*; or $c_2(< c_1)$ times STD larger, then we mark it as *slow*; otherwise, the road segment will be marked as normal. This way, the real-time traffic map can be generated.

When a road segment is marked as slow or very slow, WiLocator will further detect the root cause and identify the traffic anomalies in the following way. For each trajectory $(lat_i, long_i, t_i)(i = 1, 2, \cdots, n)$ where $(lat_i, long_i)$ denotes the bus position p_i at time stamp t_i, if there are two integers $1 < k < m < n$ such that $d_r(p_{i-1}, p_i) < \delta, k < i \leq m$ and $d_r(p_{i-1}, p_i) > \delta$ otherwise, we regard the location between p_k and p_m as the anomaly site (e.g., the location between A and B in Fig. 6) such as road construction, traffic accident, etc. The system parameter δ is determined based on the historical road distance during a scanning period on the corresponding road segment in the similar way as described above for determining parameter c_1 (or c_2). Other possible cases causing a false anomaly, such as the bus waiting for riders' boarding at a stop or traffic light turning green at an intersection, can be



(a) (b) (c)

Fig. 8: The CDF of positioning errors (a) and arrival time prediction errors (b), and the mean prediction error against the number of bus stops.

easily identified based on the bus position.

B. Experiment Results

We implement a prototype of WiLocator on Nexus 5 with the Android platform, and conduct experiments on the buses of routes 9, 14, 16 and the Rapid Line, mostly running on a main street in the Metro-Vancouver, Canada (see Fig. 7), and collect the real data of 3-week period. Each bus route shares some overlapped road segments with at least one route. Please see Table I for the detailed information about the four routes. The WiFi scanning period is set to be 10 seconds.We construct the SVD and then infer the position of a bus based on the recorded RSS readings. We obtain the geo-tags of WiFi APs from Google Map and Shaw Go WiFi[6]. During the SVD construction, the readings from unknown APs (i.e., without geo-tags) are ignored which has little impact on the results as there are at least three geo-tagged APs distributed along each road segment of the main streets, and we simply regard that all the factors affecting signal propagation are the same for APs, as the transmitted power of the WiFi APs is often limited and on-site survey is too expensive.

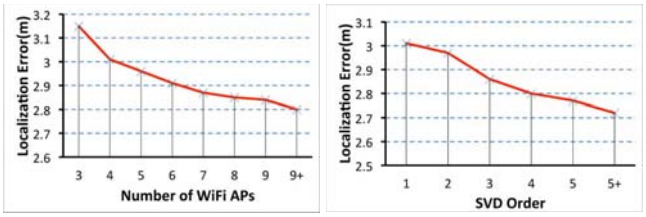
1) Bus Positioning Accuracy: Fig. 8 (a) shows the cumulative distributed function (CDF) of the positioning errors, defined as the road length between the estimated position and the real position, of each route by WiLocator. Despite the unstable WiFi signals, with sufficient WiFi APs along the road segments, WiLocator achieves a high accuracy with the median error less than 3m, showing that the SVD is promising for real-time bus positioning.

Fig. 9 (a) depicts the average positioning error against the number of WiFi APs. We observe that with the increasing of AP number, the positioning error decreases slowly from 3.15m to 2.8m. Even though these positioning results might not be very accurate for indoor settings, it is sufficient for predicting the bus arrival time with high accuracy, implying that in practical, we do not require too many WiFi APs for bus tracking. In Fig. 9 (b), the positioning error does not change significantly when the oder of SVD increases, and 2-order SVD is often enough in our tested scenarios.

[6]https://www.shaw.ca/wifi/hotspots/.

Route	# of Stops	Length(km)	Overlapped Length(km)
Rapid Line	19	13.7	13
9	65	16.3	13
14	74	20.6	16.2
16	91	18.3	9.5

TABLE I: Information of the four investigated bus routes. The overlapped length indicates the length of the overlapped road segments shared with one or more other routes.



(a) (b)

Fig. 9: The positioning error vs. the number of WiFi APs (a) and the order of SVD (b).

We also conduct an experiment on a road segment in a campus where the deployment of WiFi AP is almost as dense as in urban environments. We measure the RSS from the surrounding WiFi APs (shown in Table II) when the riding bus passes different locations (please see Fig. 10). We rank the RSS values and construct the second-order SVD based on the ranks. The error for location A, B and C is 2m, 2m and 2m respectively, and the average error is thus only 2m.

2) Arrival Time Prediction: We exploit the real data to compute the seasonal index of travel time on each road segment, based on which we divide each weekday into 5

Location	List of surrounding WiFi APs (RSS in dBm)
A	AP10(-70), AP9(-71), AP11(-79)
B	AP9(-71), AP10(-74), AP4(-76), AP5(-78), AP11(-79)
C	AP4(-50), AP5(-63), AP1(-64), AP2(-66), AP9(-78)

TABLE II: The measured RSSI values from WiFi APs at different locations.

Fig. 10: An experiment scenario. The noisy reading by GPS is mapped to the true location of bus on a one-way road segment. The WiFi APs are numbered and the lines represent the Voronoi edges of SVD generated by the APs. The locations A, B, C are ground truth of the bus position by GPS, and the estimated locations are marked by the yellow bus shapes.

time slots: <8:00AM, 8:00-10.00 AM (morning rush hours), 10:00AM-6:00PM, 6:00PM-7:00PM (afternoon rush hours), and>7:00PM. We are most concern the bus arrival time prediction during rush hours, which is rather challenging. Fig. 8 (b) gives the CDFs of the arrival time prediction error by WiLocator and the Transit Agency. We observe that the errors by WiLocator and the Transit Agency are comparable, except that the Transit Agency produces the maximal prediction error about 800 seconds during the rush hours, while the maximum error by WiLocator is 500 seconds. Fig. 8 (c) presents the relationship between the average arrival time prediction error and the number of bus stops in rush hours [7]. Note that for the Rapid line, pairwise stops are separated farther away than other routes, thus it suffers less from the traffic jam in the overlapped segments, and achieves the lowest prediction error. As the farther away the bus stop, the more uncertainty, we find an increasing trend of the prediction error, but overall the results are acceptable during rush hours, with the maximum error of 210 seconds.

3) Traffic Map Generation: First, for each bus route R_j on the road segment e_i, at time slot l, we compute the historical travel time residual $\hat{\epsilon}_{i,l} = \frac{\sum_j (T_h(i,j) - T_r(i,j))}{J}$, where J is the number of buses having passed segment e_i, to exclude the impact of route-related factors, and compute the standard variation of the residual $\sigma(\epsilon_{i,l})$. Then, when a bus of route R_j has just travelled a road segment, say e_i, we compute the statistic $z_{ij} = \frac{\epsilon_{i,l} - E(\epsilon_{i,l})}{\sigma(\epsilon_{i,l})}$. Based on the Rule of Thumb, if $z_{ij} < -1.64$, we mark the segment e_i as "very slow" with 95% confidence, according to the rule of thumb; if $z_{ij} < -1.00$, we mark the segment e_i as "slow", otherwise it's normal.

Fig. 11 shows the traffic map by WiLocator, Transit Agency and the Google Map. As can be seen, the traffic map by the

[7]For comparison, here we only show the prediction errors of the first 19 stops since the Rapid line has only 19 stops.

Fig. 11: The traffic maps during a rush hour on W Broadway by WiLocator (a), the Transit agency (b) and Google Map (c).

Transit Agency has some unconfirmed segments, which is to our surprise, and the Google Map detects some true anomalies, such as the accident (leftmost) and traffic jam (shown in red segments). We find that after zooming in, there are also unmarked road segments by Google Map. WiLocator also detects some anomalies, but none road segment is unmarked as it exploits the temporal constancy to infer the future traffic. As can be expected, with more historical data, WiLocator can offer a more accurate real-time traffic map.

VI. Related Work

A. WiFi-Based Localization

An active approach of this line is RF Fingerprinting-based [3], [9], [10], [10], [11], [17], [24], [25], which require an offline training phase and online matching phase. During the offline training phase (so-called calibration), an on-site survey of RSS measurements is conducted to build a fingerprint database, such that a mobile device can be localized by matching the observed RSS measurements against the fingerprint database. As the calibration is labor-extensive and time consuming, a body of solutions are proposed to reduce the effort, such as [8], [10], [14], [22], [23], albeit at the cost of lower accuracy. Some researchers also propose RF propagation model based schemes. For instance, Assuming a dense coverage of WiFi, EZ [6] models the constraints of the reported observation by physics of signal propagation and locate the mobile user based on the proposed genetic algorithm. Again, solutions of this line suffer from low accuracy.

B. Real-time Bus Tracking

In literature, there are two major approaches to track the bus. One is GPS-based, and the other relies on cellular infrastructure. The major limitations of GPS-enabled AVL based tracking system are that they are extremely power-hungry, costly and suffer from urban canyons. Biagioni *et al.* [4]

presented an energy-friendly system, named EasyTracker, for automatic bus tracking by using COTS smartphones. However, the inherent nature stemmed from GPS makes it work poorly in urban environments. An alternative approach is smartphone-based crowd-sensing via cell-ID sequence matching [15], [27]–[29]. However, in cities, the coverage of a cell tower can reach 800m around, and it take several minutes for the bus rider to capture a stable cell-ID sequence, potentially incurring a low accuracy. Also the overlapped road segments in urban environments pose great challenges for real-time bus tracking.

VII. Conclusions and Future Work

We study the issue of real-time bus tracking and arrival time prediction in urban areas with smartphone-based crowd-sensing, and design a scheme to leverage the mobility constraint of a bus, and the travel time consistency of buses on the same road segment. We instantly track a bus by using the scanned WiFi information available for bus riders to construct a signal Voronoi diagram based on the rank of RSS measurements from nearby WiFi APs, instead of relying on fingerprinting the WiFi APs or signal model. Also for each road segment we compute the seasonal index of the travel time, and thereon the historical travel time of each time slot. By incorporating the historical and most recently travel time of the bus routes shared the same road segment, we obtain a predicted arrival time of the bus route.

WiLocator is by no means exclusive; it can seemly integrate with GPS or Cell-ID based location systems. For instance, when a smartphone scans no WiFi information for a while, the GPS module is activated so that the system can adaptively work from WiFi-coverage areas to GPS viable environments. Other built-in sensors, such as accelerator, etc., can be leveraged to improve WiLocator's performance. In addition to the applications for bus arrival time prediction and traffic map generation, we also envision that the proposed SVD has the potential for facilitating navigation in urban environments where an inaccurate positioning of the vehicle might lead to a wrong turn instruction. These will be our future work.

VIII. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61202460, Grant 61572219 and Grant 61271226; by the China Postdoctoral Science Foundation under Grant 2014M552044; by the Thousand Talents Plan under Grant 61571202. The corresponding author of this paper is Hongbo Jiang.

References

[1] F. Aurenhammer. Voronoi diagrams−a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

[2] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: Mobile phone localization via ambience fingerprinting. In *Proc. of ACM MOBICOM*, pages 261–272, 2009.

[3] P. Bahl and V. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proc. of IEEE INFOCOM*, volume 2, pages 775–784, 2000.

[4] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson. EasyTracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones. In *Proc. of ACM SenSys*, pages 68–81, 2011.

[5] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy characterization for metropolitan-scale Wi-Fi localization. In *Proc. of ACM MOBISYS*, pages 233–245, 2005.

[6] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proc. of ACM MOBICOM*, pages 173–184, 2010.

[7] I. Constandache, S. Gaonkar, M. Sayler, R. Choudhury, and L. Cox. EnLoc: Energy-efficient localization for mobile phones. In *Proc. of IEEE INFOCOM*, pages 2716–2720, 2009.

[8] A. Goswami, L. E. Ortiz, and S. R. Das. WiGEM: A learning-based approach for indoor localization. In *Proc. of the Seventh COnference on Emerging Networking EXperiments and Technologies (CoNEXT)*, pages 3:1–3:12, 2011.

[9] Y. Gwon, R. Jain, and T. Kawahara. Robust indoor location estimation of stationary and mobile users. In *Prof. of IEEE INFOCOM*, pages 1032–1043, 2004.

[10] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proc. of ACM MOBICOM*, pages 70–84, 2004.

[11] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless ethernet. In *Proc. of ACM MOBICOM*, pages 227–238, 2002.

[12] M. Lee and D. Han. Voronoi tessellation based interpolation method for Wi-Fi radio map construction. *IEEE Communications Letters*, 16(3):404–407, 2012.

[13] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-accuracy trade-off for continuous mobile device location. In *Proc. of ACM MOBISYS*, pages 285–298, 2010.

[14] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *Proc. of ACM MOBISYS*, pages 299–314, 2010.

[15] 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*, pages 293–306, 2011.

[16] J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an organic indoor location system. In *Proc. of ACM MOBISYS*, pages 271–284, 2010.

[17] T. Roos, P. Myllymaki, and H. Tirri. A statistical modeling approach to location estimation. *IEEE Transactions on Mobile Computing*, 1(1):59–69, 2002.

[18] N. Swangmuang and P. Krishnamurthy. Location fingerprint analyses toward efficient indoor positioning. In *Proc. of IEEE PerCom*, pages 100–109, 2008.

[19] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. In *Proc. of ACM SenSys*, pages 85–98, 2010.

[20] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Proc. of ACM SenSys*, pages 85–98, 2009.

[21] A. Thiagarajan, L. S. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod. Accurate, low-energy trajectory mapping for mobile devices. In *Proc. of USENIX NSDI*, pages 267–280, 2011.

[22] C. Wu, Z. Yang, Y. Liu, and W. Xi. WILL: Wireless indoor localization without site survey. In *Proc. of IEEE INFOCOM*, pages 64–72, 2012.

[23] C. Wu, Z. Yang, Y. Liu, and W. Xi. WILL: Wireless indoor localization without site survey. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 24(4):839–848, 2013.

[24] M. Youssef and A. Agrawala. Handling samples correlation in the Horus system. In *Proc. of IEEE INFOCOM*, volume 2, pages 1023–1031, 2004.

[25] M. Youssef and A. Agrawala. The Horus WLAN location determination system. In *Proc. of ACM MOBISYS*, pages 205–218, 2005.

[26] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng. I am the antenna: Accurate outdoor AP location using smartphones. In *Proc. of ACM MOBICOM*, pages 109–120, 2011.

[27] P. Zhou, S. Jiang, and M. Li. Urban traffic monitoring with the help of bus riders. In *Proc. of IEEE ICDCS*, pages 21–30, 2015.

[28] 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*, pages 379–392, 2012.

[29] P. Zhou, Y. Zheng, and M. Li. How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing. *IEEE Transactions on Mobile Computing*, 13(6):1228–1241, 2014.