# Highway Vehicle Trajectory Reconstruction Using Sparse and Noisy Communication Base Station Fingerprints

Yingbing Li, Yan Zhang, and Min Chen

Abstract— **The current application of highway toll system, generally use Dijkstra algorithm to calculate the shortest path of vehicles from the entrance to the exit to charge. This means that managers have no way of knowing the exact trajectory of vehicles. And different routes of highways are often funded and operated by different investors. To address this problem, this paper presents a new algorithm to reconstruct trajectories from sparse and noisy fingerprint signals from communication base stations (CBSIDs), with practical application in a high-speed toll collection system in Hubei Province, China. In this solution, we designed an inexpensive Internet of Things (IOT) device that collects signal fingerprint identification numbers from CBSIDs at a low sampling rate. These CBSIDs are then matched with a special CBSID-anchor radiomap, converting the sequence of CBSIDs into a sequence of candidate anchors (toll stations and intersections on highways). Finally, a trajectory mapping algorithm is run to process these candidate anchors and to generate the complete driving route. In the experiment on both simulated and field routes, results show that the proposed algorithm can effectively reconstruct the driving routes of vehicles and achieve intelligent sensing and recognition of vehicle movement routes. The upgraded toll collection system meets the needs of efficient motorway investment, maintenance and management.**

Index Terms— **Communication base station, fingerprint matching, map matching, Trajectory Reconstruction , IOT Technology**
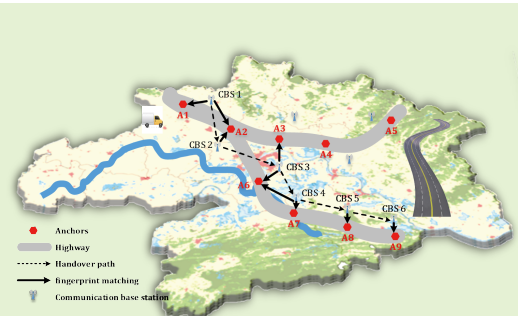
I. INTRODUCTION

TOLL collection is a fair and efficient strategy for highway investment, maintenance, and management, and is widely adopted by countries in Asia, Europe, and North America [1]– [3]. Existing highway toll collection systems generally only record the entrance and exit on the highway, then calculate the toll according to the shortest path between them [4], [5]. However, such a strategy may incur a discrepancy between the traveled distance and the tolled distance. 1. Different routes of the highway are built by different investors, and as the highway network becomes more complex, there is likely to be more than one route between the same entrance and exit, which poses a problem for profit distribution [6]. 2. Since the existing system cannot record the actual trajectory of vehicles. Drivers can drop off passengers or goods at service areas closer

Yingbing Li is with School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China (e-mail: ybli@whu.edu.cn).

Yan Zhang is with the State Key Laboratory of surveying, mapping and remote sensing information engineering, Wuhan University, Wuhan 430079, China and the College of Design and Engineering, National University of Singapore, 117566,Singapore (e-mail: sggzhang@whu.edu.cn).

Min Chen is with the School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China (e-mail: germainechen@whu.edu.cn).

to their destinations and drive vehicles back to their origins to save passage costs. This system vulnerability brings millions of economic losses every year. In this paper, we propose a new and economically reliable trajectory reconstruction method to solve these two problems.

Here, we present some concrete statistics about drivers' choice of highway routes. The dataset consists of real driving routes on highways collected in 2018. Each route is made up of cellular localization positions sampled at an average interval of 6s. A total of 67,466 routes were obtained with a total length of 17,267,864.07 km. Through comparison with the shortest route between the same entrance-exit, we found that 2914 routes are different from their shortest counterparts, and the total length of them is 551,497.96 km longer. The total tolled distance is 3.19% shorter than the real distance. The 67,466 routes are made up of 19497 distinguishing entrance-exits, and between 4054 (accounting for 20.32%) of the entrance-exits, drivers have chosen an alternative route other than the shortest path. Fig. 1 shows the 20 most frequently visited entrance-exits. In 9 out of the 20 cases, drivers have chosen alternative routes other than the shortest one. The total length of routes taken by drivers between points 146 and 165, for example, is approximately 4,939 km longer than that of the shortest route. As for the rest 11 entrance-exits, it is apparent from Fig. 2 that the routes between them are fairly short, and that is

why drivers have not taken different routes. When the distance and the complexity of the road network increase, diversity of actual driving routes increases. Considering that highways are commonly invested by a number of companies and their charging standards vary from each other, the discrepancy between the paid toll and the true toll will be more significant.
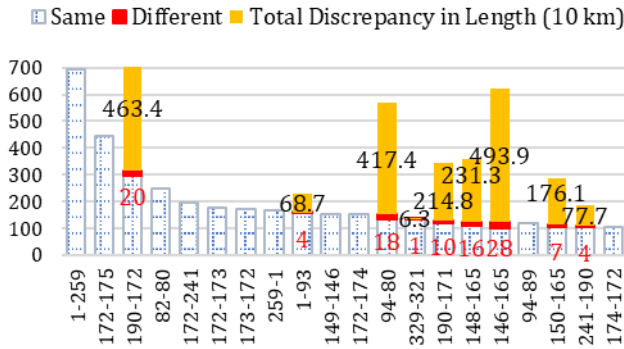

Fig. 1. Discrepancies between the shortest route and the real route of the 20 most frequently visited entrance-exits. The vertical axis shows the number of routes; the horizontal axis represents entrance-exits.

The statistics convincingly prove that the shortest-path strategy may cause significant loss in interest for investors and shareholders. In order to toll accurately, there is an urgent demand for developing a more effective strategy for tracking drivers' actual driving routes.

There are a variety of solutions to the vehicle route tracking issue, such as Global Navigation Satellite System (GNSS)& 5G, Radio Frequency Identification (RFID) with License Plate Recognition (LPR), and fingerprint-based methods [7]–[10]. These methods should be carefully examined in the context of highway toll collection, in terms of availability, energy-efficiency, infrastructure investment, and latency.

GNSS, for example, is the default solution to most outdoor localization applications [11]. Large number of applications has been developed based on GNSS to tackle the trajectory tracking issue. These applications range from EasyTracker [12] and StarTrack [13] in the academic realm to everyday mobile navigation applications such as Baidu Map and Google Maps. However, there are also some limitations in GNSS-based applications, such as high energy consumption, poor performance in urban canyons, and investment in signal receiving devices [14]–[16]. Hence, GNSS may not be the optimal choice when it comes to the application in toll collection systems where energy efficiency and investment must be taken into careful consideration [17], [18].

RFID with PLR has been used for vehicle access control [19] and parking management [20], [21], and shows good potential in Electronic Toll Collection (ETC) applications, since it allows tolling without stopping the vehicle [22]. Nevertheless, the implementation of a network-based LPR system is often accompanied by a considerable investment of infrastructure and high complexity in computation [23], which makes the wide-spread application difficult.

The principle of the fingerprint-based localization method is to connect vision, motion, or signal fingerprints with a set of location-tagged signatures such as landmarks, directions, distances, and coordinates [24]. Vision fingerprint-based localization methods use images or videos to locate the points of interest [25]. Although crowd-sourcing and advanced image processing technologies have improved the localization accuracy and decreased the latency time of such methods, the inherent overhead and computational complexity are still non-trivial [26], [27]. Motion fingerprints are generated by motion sensors like accelerometer, gyroscope, and electric compass embedded in smart-phones [28]. They are often combined with other methods to improve the localization performance because the inherent noise in the collected data often causes huge errors in location estimates [29]–[31]. Wireless-Fidelity (Wi-Fi), Bluetooth, and ZigBee are popular signals used in indoor localization [32]. Wi-Fi-based localization method has also been used in outdoor urban areas when enough Access Points (AP) are around [33]. But the availability of Wi-Fi and Bluetooth signals is often highly limited in some rural areas along the highways. On the contrary, signals from communication base stations (CBSs), called CBS fingerprints thereafter, are almost ubiquitous along the highways, which can be used for localization without the need for additional infrastructure. In addition, obtaining CBS fingerprints consumes less energy compared to GNSS [34]–[36], and the processing of CBS fingerprints is simpler and causes less latency than that of vision fingerprints. In view of availability, energy-efficiency, infrastructure investment, and latency, CBS fingerprint is more appropriate in the context of highway route tracking application.

In this paper, we propose a CBS fingerprint-based solution to the highway route tracking problem. The solution designs a low-cost signal receiving terminal that is distributed to each vehicle at the entrance of the highway. The device is recovered at the highway exit and automatically reconstructs the trajectory and generates toll orders based on the information recorded by the device. It does not require drivers to install additional APPs, periodically collects the identities of registered CBSs (CBSIDs) and then rebuilds vehicle trajectories from these sparsely sampled sequences of CBSIDs.

1)The method is simple and efficient. Compared to previous works concerning CBS fingerprint map matching [31], [34], [37], the method requires limited information about the fingerprint (only CBSID of the register CBS). It associates the CBSIDs with special anchors on highways, which helps to capture vehicles' mobility and improve trajectory mapping accuracy. In the experiment on 197 simulated routes, the method accurately recognizes 99%, 99%, 99%, 99%, 97.5%, 96.4%, and 93.4%, of the routes at the sampling intervals of 5 s, 30 s, 1 min, 5 min, 10 min, 15 min, and 20 min, respectively. In the experiment on 3 field routes, CBSIDs are collected every 15 min. The method accurately identifies two of the routes, and the precision and recall of the other route are 0.92 and 1, respectively. The method shows strong robustness against sparseness and noise of data.

2)The method has been widely used in the highway toll collection system of Hubei Province now. A toll collection system based on the method not only tracks a vehicle's real driving route so that toll computation can be performed accu-
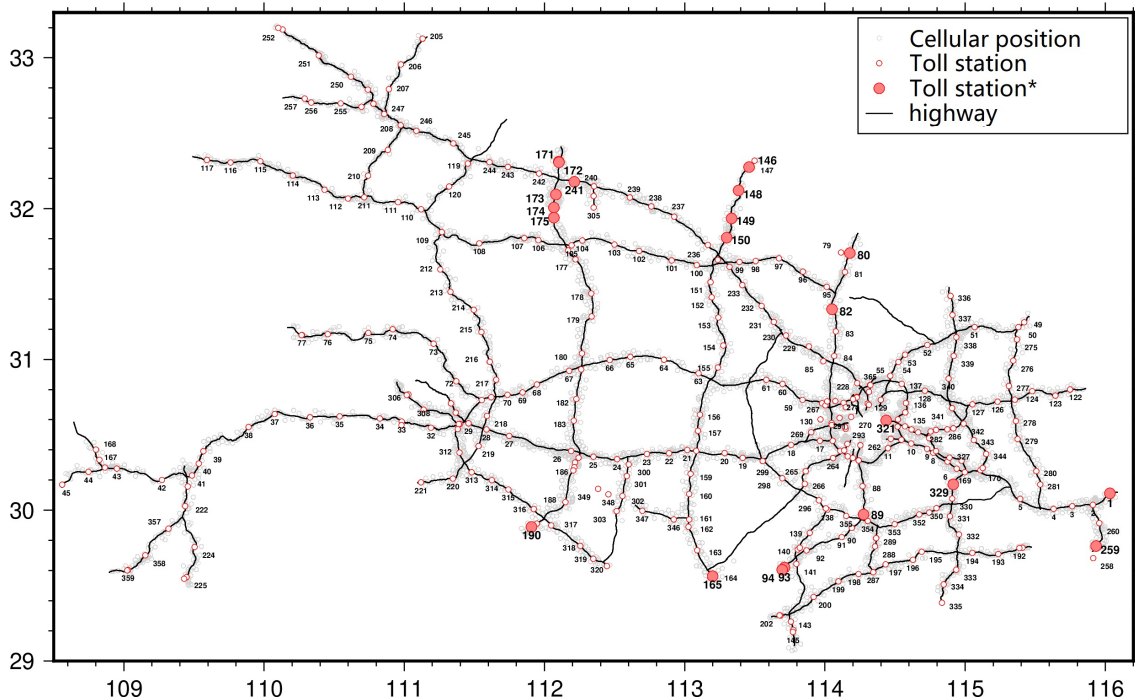
Fig. 2. Distribution of toll stations and sampled cellular positions. "Toll station*" represents the most frequently visited starting or ending toll station.

rately, but also is quickly responsive, highly available, energy-efficient, and infrastructure-installation-free. The rest of the paper is organized as follows: In Section II, the related works are summarized; Section III introduces the proposed method of route recognition based on CBS fingerprints; Section IV verifies the method by field experiments in Hubei; finally, Section V concludes the present work with a deeper discussion on the proposed method.

II. RELATED WORKS

A. CBS Fingerprint Matching

Fingerprint matching is an indispensable module of a fingerprint localization system because it builds the connection between fingerprints and location-tagged units in the real world [24]. There are two kinds of methods for fingerprint matching: one directly converts fingerprints to coordinates [38]–[40]; the other incorporates sequential information into the mapping process [34], [37], [41].

Methods of the first kind match the target fingerprint against fingerprints in the radiomap, then return the coordinates of its matches. In such cases where there are more than one matched fingerprints, the location is computed by the average of the K nearest neighbors (KNN) based on Euclidean Distance [42] or estimated by probabilistic methods [43]. Cell Identity (Cell-ID) and Received Signal Strength (RSS) are most frequently used matching features. C. Ang et al. [33] compared the two criteria and concluded that although the incorporation of RSS decreases the granularity of the matching algorithm, the simple Cell-ID matching scheme is less sensitive to the fluctuation of signal strength and thus shows higher robustness.

Studies in [34] and [44] point out that sequencing fingerprints before matching can improve localization accuracy

because the process of sequencing incorporates spatial and temporal constraints into the subsequent mapping procedure, which naturally reduces the probability of a mismatch. Research works in [31], [34], [41] utilize Hidden Markov Model (HMM) to simulate a vehicle's transition from one spatial unit to the next, thus incorporating road constraints and temporal information into the fingerprint matching process. Experimental results have proved the effectiveness of HMM in fingerprint matching. However, HMM requires thorough training of the model, and the performance of one node is highly dependent on that of its preceding node. One mismatched node may transfer the cascading effect to all the nodes behind it, which may decrease the matching accuracy of CBS fingerprints that are sparsely sampled.

The Smith-Waterman algorithm has also been used in fingerprint sequence-matching problems [45], [46]. It is an effective method for local sequence alignment [47]. By weighting each fingerprint in the sequence, the algorithm produces a sum of fingerprint weights and uses it as the matching score. The algorithm is well suited for regular routes, such as bus routes and subway lines. But it requires a continuous collection of fingerprints and a comprehensive offline war-driving for radiomap construction.

In order to enhance the robustness against signal strength fluctuation and to simplify the matching process, the unique identifier of CBS (CBSID) is used to search for matches. CBS fingerprints are not directly converted to coordinates. Instead, they are connected to anchors, i.e., toll stations and intersections, where critical driving actions take place so that spatial constraints are incorporated. Each fingerprint is independently matched with the radiomap hence it is not subject to the interference of preceding nodes. In addition, a weighting scheme is adopted for fingerprint matching in reference to

the Smith-Waterman algorithm. The above settings enhance the effectiveness of the method in tackling sparse fingerprint sequences.

B. Map Matching Based on Sparse and Noisy CBS Fingerprints.

Cellular network localization data are very noisy. If only the associated CBS information is used, the mean error of localization is 2 km [48]. And fingerprints sometimes can only be collected in a sparse time interval due to limitations on energy and data storage. The noise and sparseness of CBS fingerprints make map matching of the real route a challenging problem. Newson et al. [49] simulated the situation by adding noise to GNSS data and found that with 100 m noise standard deviation and 9 min sampling interval, the fraction of mismatched route almost reaches 0.9.

Newson et al. [49] proposed a HMM method to tackle the problem, and it has been further adopted and improved by later works, such as C-Track [34], WheelLoc [31], and SnapNet [48]. It is recommended to refer to the work of Dalla Torre et al. [50] that gives a thorough and detailed survey of these HMM methods applied in CBS fingerprint map matching. Although HMM has shown good performance, there are still some problems. The authors of [41] and [51] argue that mobility in a network is actually non-Markovian, and a unified transition probability can lead to mismatches of road segments. The authors hence incorporated the statistics of driver's behavior to improve the map matching accuracy.

Apart from HMM-based methods, there are other solutions to map match sparse and noisy CBS fingerprints. For instance, Gunnar et al. proposed a special searching corridor to locate possible road segments and achieved good performance using only the CBSID [52].

We provide a simple and efficient map matching solution to the problem. The algorithm considers the specific mobility patterns of vehicles on the highway and converts into a certain trajectory the sequence of register CBSIDs.

III. METHODOLOGY

A. Problem Statement

The present work is inspired by the "handover" of CBSs that takes place in the mobile communication network. As shown in Fig. 3, as a vehicle moves within a network, a communication device equipped in the vehicle chooses different CBSs as its serving stations to keep the continuity of communication [53]. Handover between CBSs happens when the signal strength from the current serving CBS falls to a certain extent. During this process, two paths are simultaneously generated: 1) the vehicle's driving path in the real world; 2) the pseudo-path concatenated by CBSs. It is clear that the two paths are spatially close to each other. The spatial proximity enlightens an idea: can we use the sequence of CBSs to determine the unknown vehicle's driving path?

To address the problem, the previous works listed in Subsection B, Section II have proposed different approaches. In general, there are two challenges in inferring the driving route from CBS fingerprints. The first challenge is that the CBS
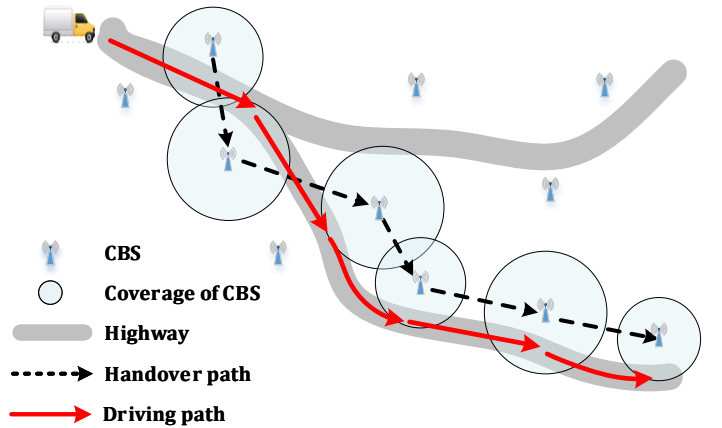

Fig. 3. The pseudo path of CBSs and its relationship with the vehicle's driving route on the highway.

fingerprint does not explicitly indicate the location of the vehicle. All we can do is to infer the possible location of the vehicle, i.e., the overlap of the circle and the highway in Fig. 3. Since the locations of CBSs are unknown and their signal coverages vary from each other, it is difficult to find the overlap. The localization uncertainty of each fingerprint increases the uncertainty in inferring the whole route, and this is the second challenge we face. We must develop an effective method to extract the real driving route from all possible locations. The problem is defined as:

$$Real\left(route|f_1, f_2, \cdots, f_n\right) = MaxLikelihood_\Phi\left(route_x|f_1, f_2, \cdots, f_n\right)$$
(1)

where $\Phi = \{p_1, p_2, \cdots, p_m\}$, $route_x \in \Phi$, f_1 is the sequence of CBS fingerprints; Φ is the set of all possible locations of the vehicle; and routex is a subset of Φ.

B. System Overview

We propose a highway vehicle-route-recognition system to meet these challenges (Fig. 4). The lower half of the figure illustrates the process of fingerprint collection. As soon as a vehicle enters the highway, the CBSIDs of serving stations are periodically recorded until the vehicle arrives at the exit. Then the recorded sequence of CBSIDs together with the information of the enter and exit is uploaded to the route-recognition module shown in the upper part of Fig. 4.

The route-recognition module converts the sequences of CBSIDs into the vehicle's driving route. It includes two key modules: the fingerprint matching module and the trajectory mapping module. The former module transfers the sequence of CBSIDs into a sequence of anchors, i.e., toll stations and intersections on roads; the latter module then converts these anchors into a complete route on the road network. The fingerprint matching module requires a pre-constructed radiomap to find matching anchors; and the trajectory mapping module needs the information of road networks. Hence, an offline war-driving is also required to build the radiomap and updates the road network database. Detailed descriptions of the two modules are given in subsections C and D.
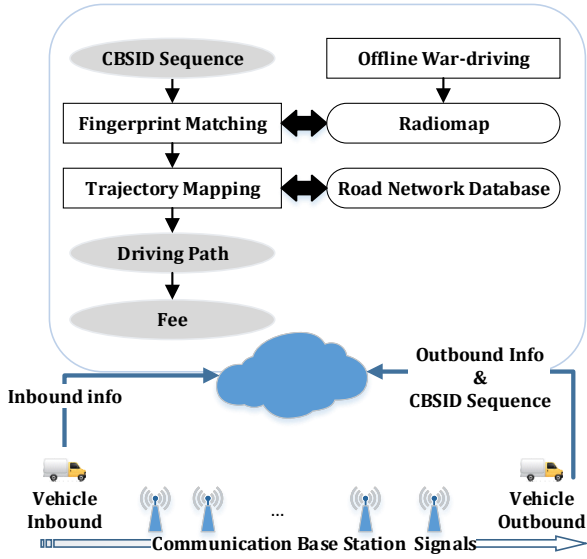
Fig. 4. The architecture of the highway driving-route-recognition system.

C. Fingerprint Matching

Fig. 5 shows the workflow of the fingerprint matching module. First, an offline war-driving is conducted to collect CBSIDs along highways. The spatial unit refers to the location range of the vehicle, within which several CBSIDs are recorded. Then, the radiomap connects each spatial unit with a set of CBSIDs. Lastly, given a CBSID, the location of the vehicle can be inferred from the radiomap. For example, when the vehicle moves in the range of spatial unit 1, it receives CBSIDs 1 and 2. So, with CBSID 2 the possible locations of the vehicle should be spatial units 1 and 2 according to the radiomap.
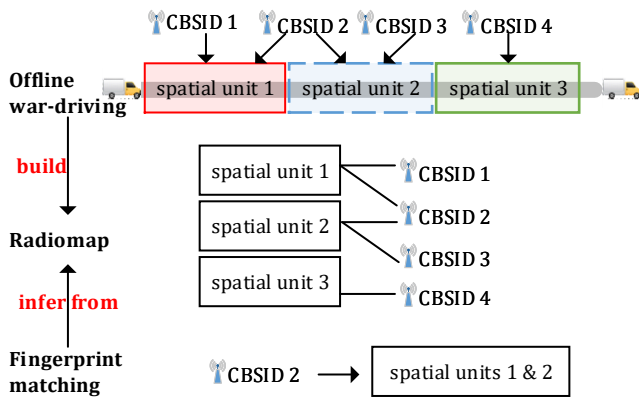

Fig. 5. Workflow of the fingerprint matching module.

1) Offline war-driving: Fig. 6 shows how CBS fingerprints are collected in offline war-driving. The vehicle moves along the highway and collects data samples at short intervals (five second). Each data sample is composed of the time of sampling, GNSS coordinate of the vehicle, and CBSIDs of one register CBS and at most six neighboring CBSs. Sampling point, such as P1, refers to the location of the vehicle when it

records a data sample, and the GNSS coordinate is assumed precise enough to represent the sampling point. Anchors are toll stations and intersections on the highway, and they are manually recorded.
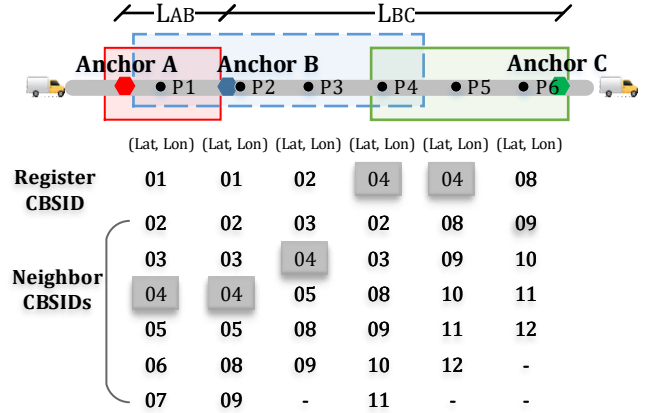

Fig. 6. Example of data samples and the illustration of fingerprint clustering, taking anchors as the clustering kernel. Black dots represent the sampling points of CBS fingerprints (The input is the CBSID information received at the sample point and sample location P_i. The output is the CBSID-anchor radiomap which is the correspondence between CBSID and anchor, one CBSID may correspond to zero, one or more anchors. The detailed radiomap construction process is summarized as Algorithm 1).

2) Spatial unit: Figure 7 illustrates three types of radiomaps in which the spatial units are coordinates, grids, and road segments, respectively. The figure also shows the workflow of fingerprint matching corresponding to the three radiomaps.
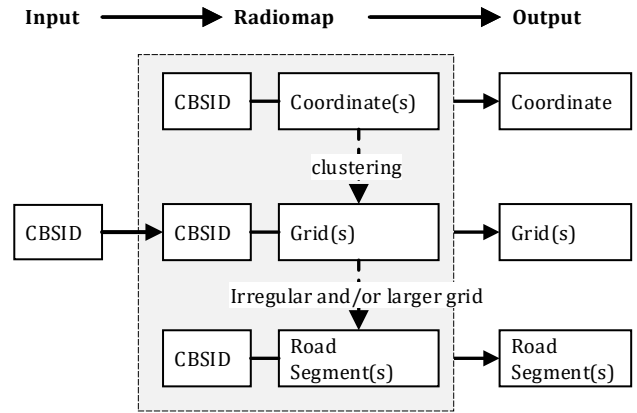

Fig. 7. Three types of radiomap and the workflow of CBSID fingerprint matching (The granularity of road segment positioning is the coarsest, but in the context of Hubei province's total highway mileage of nearly 8,000 km, it is more appropriate to choose road segments as the spatial unit. It avoids the complicated step of mapping communication base stations to roads and is computationally efficient, as a CBSID is often matched to only a few anchors, which improves the efficiency of the next trajectory reconstruction. In fact, there are nearly 10,000 communication base stations in our study area in total, while there are only hundreds of anchors).

The coordinate-based radiomap converts fingerprints into precise coordinates, so it has the smallest localization granularity. The CBSID-grid radiomap divides the area of interest into regular grids and connect these grids with CBSIDs.

We treat the CBSID-road segment radiomap as a special type of CBSID-grid radiomap because road segments can be seen as grids on the road with irregular (and often longer) lengths. Notwithstanding the coarse granularity of localization, choosing road segment as the spatial unit is more appropriate in the context of highway application. The reason is twofold. On the one hand, the coordinates and grids are not necessarily on the road, requiring additional work to "drag" them to the road network; whereas the road segment-based method can directly map the CBS fingerprint to the road network. On the other hand, only the CBSID is used in fingerprint matching and RSS information is not considered, so, given a single CBSID, there may be a large number of matching coordinates or grids, but only a few road segments. Discriminating the right ones from several road segments is easier than from tens of or hundreds of grids or coordinates.

Moreover, we notice that there are some unique driving patterns on highways. Restricted by traffic rules, vehicles usually go straight along the highway, and actions such as entering and exiting the highway, taking U-turns, as well as transiting between two roads are only permitted at specific sites like toll stations and intersections (anchors) [54]. These actions are vital hints for the inference of the actual driving route. In addition, anchors naturally divided highways into road segments, and it is easier to build topological relationships between points than lines, hence converting road segments into anchors can lower the complexity of computing. In consideration of the above reasons, we choose anchors on highways as kernels to cluster sampling points and generate irregular road segments. Taking anchors as the final fingerprint matching unit helps capture critical driving actions and hence improve trajectory mapping accuracy.

3) Construction of the CBSID-anchor radiomap: We use the spatial proximity and the signal coverage of CBSs to determine the affiliation of sampling points to anchors. For example, in Fig. 6, sampling point P1 is closer to anchor A than anchor B, but the distance between anchors A and B is so near that they most likely sit in the signal coverage of the same CBSs. Thus, it is reasonable to affiliate P1 to both of the anchors, so is the CBSID collected at P1. In another case, if the distance between anchors, like B and C, is fairly large, the spatial proximity is mainly considered and the sampling points are assigned to the closer anchor. As shown in Fig. 6, sampling points P4, P5, and P6 that fall in the green block are all affiliated to anchor C. The width of the green block is set to be λL_{BC}, in which the parameter λ determines the range of coverage of an anchor. If $\lambda > 0.5$, there will be an overlapping and sampling points are simultaneously assigned to both anchors.

We set the parameter *threshold* to discriminate cases AB and BC. The value of *threshold* is set to be the statistical maximum coverage of CBSs on the highway. It can be seen in Fig. 6 that the register CBS and the neighboring CBSs vary with the sampling point. For instance, CBS 04 serves as the neighboring station at sampling points P1, P2, and P3; then, serves as the register station at P4 and P5; finally, it is replaced by CBS 08 at P6 and disappears from the list. The coverage of a CBS on the highway is set to the accumulation of Euclidean distance between sampling points. The coverage of CBS 04,

for example, is the accumulated Euclidean distance between P1, P2, P3, P4, and P5, since both register and neighboring stations are counted to populate the radiomap.

The detailed radiomap construction process is summarized as Algorithm 1.

Algorithm 1: CBSID-anchor radiomap construction

Input: the sequence of anchors:
$A = (A_1, A_2, \ldots, A_n)$;
the sequence of sampling points:
$P = (P_1, P_2, \ldots, P_m)$;
the sequence of CBSIDs of register CBSs:
$C = (C_1, C_2, \ldots, C_m)$;
the Euclidean distance between two points:
$L(a, b)$;
the statistical maximum coverage of CBSs:
threshold;
the ratio of anchor coverage: λ.

Output: affiliation of CBSID to anchors.

1 **clustering of sampling points, taking anchors as the clustering kernel:**
2 **for** $i \leftarrow 1$ **to** n **do**
3   Find the sampling points $P_1, P_2, \ldots, P_k$ between A_i and A_{i+1};
4   **if** $L(A_i, A_{i+1}) \leq$ threshold **then**
5   affiliate $C_1, C_2, \ldots, C_k$ **to** both A_i and A_{i+1};
6   **else**
7   **for** $j \leftarrow 1$ **to** k **do**
8   **if** $L(A_i, P_j) \leq \lambda \, (A_i, A_{i+1})$ **then**
9   affiliate C_j **to** A_i
10   **end**
11   **if** $L(A_{i+1}, P_j) \leq \lambda \, L(A_i, A_{i+1})$ **then**
12   affiliate C_j **to** A_{i+1}
13   **end**
14   **end**
15   **end**
16 **end**

---

4) Fingerprint matching with the radiomap: It is time-consuming to traverse the whole radiomap for each CBSID of the fingerprint sequence. Supposing there are M anchors and N CBSIDs in the radiomap, the complexity of traversing the whole CBSID-anchor radiomap would be O(MN). It is observed that a CBSID includes a Location Area Code (LAC) and a Cell Identity (CID), and a single LAC may be shared by many CBSs. Therefore, searching the LAC first can narrow the targets. We put forward a two-step searching strategy that, for each CBSID, a LAC-anchor database is first searched then the anchor-CID database. For example, for CBSID 2872117361, the LAC is 28721, and the CID 17361. The LAC is firstly compared with the LAC-anchor radiomap, which outputs several anchors, such as anchors A and B. Then, we just need to compare CID with the anchors A and B in the anchor-CID database. Given N_1 LACs and N_2 CIDs, the complexity of the searching algorithm would be $O\left(M\left(N1 + N2\right)\right)$. $N = N_1 N_2$, $N_1 + N_2 \ll N$. The two-step searching strategy makes it possible to quickly match candidate anchors for the

next step of trajectory mapping.
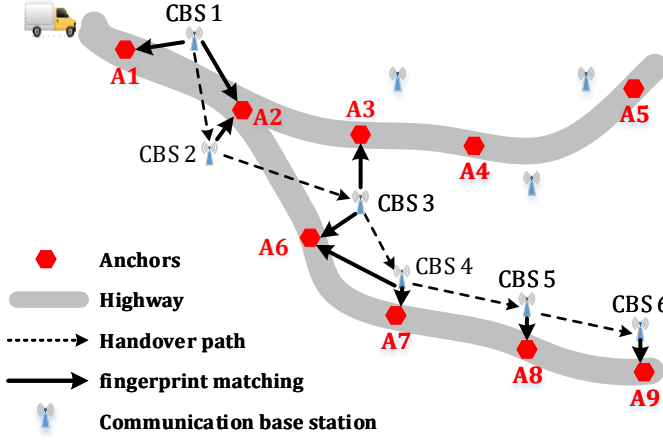
D. Trajectory Mapping

Fig. 8. Illustration of fingerprint matching results.

Fig. 8 depicts the results of fingerprint matching. All anchors are renumbered along the highways in the direction from north to south and from east to west. Given the sequence of CBSIDs: {CBSID1, CBSID2, CBSID3, CBSID4, CBSID5, CBSID6}, the output of the fingerprint matching module is: {{A1, A2}, {A2}, {A3, A6}, {A6, A7}, {A8}, {A9}}. The aim of trajectory mapping is to convert the sequence of candidate anchors to a continuous trajectory. A rule-based trajectory mapping algorithm is put forward to filter and sequence these candidates. There are four main steps for this algorithm: 1) weighting; 2) concatenation; 3) filtering; and 4) generation of the complete driving route.

1) Weighting of the anchors: Fig. 9 illustrates how anchors are weighted. Anchors are categorized into three types, and each type is given a different weight. Entering and exiting anchors (A1 and A9) are given the highest weight (e.g., 10), as they are explicitly written into the record. If a CBSID is associated with only one anchor (such as A2, A8, and A9), the anchor is also given a high weight (e.g., 5) because of the high possibility of the vehicle's coming close to the anchor. If the anchor is an intersection, such as A2, it also gets a high weight as it marks the transition between roads. Other anchors that do not belong to the above types are given a low weight (e.g., A3, A6, and A7).

2) Concatenation of weighted anchors: Fig. 9 also depicts the concatenation of weighted anchors. For each candidate anchor, if there is no identical record in the previous sequence of anchors, it is directly affiliated to the end of the sequence. Otherwise, the anchor's weight is added to its identical record. For instance, A2 is in the lists of the candidate anchors of both CBSID 1 and CBSID 2, and the weight of A2 is accumulated accordingly. The output trajectory of this step is {A1, A2, A3, A6, A7, A8, A9}.

3) Filtering: A high-pass filtering is used to filter out low-weight candidates. Although this may result in mistakenly discarding the correct anchors (for example, A7 in Fig. 9), the strategy avoids more severe interference of noise caused
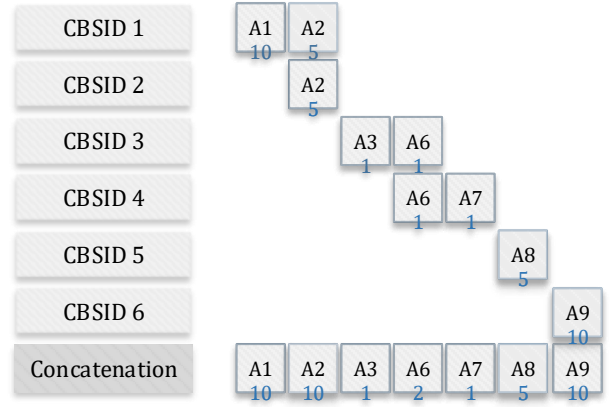

Fig. 9. Weighting and concatenation of candidate anchors.

by incorrectly matched anchors. It is believed that these high-weight anchors are more likely to match the real route (Eq. (2) is the weighting rule for anchor in different cases). The output route is generated with these high-weighted anchors: {A1, A2, A6, A8, A9}.

$$Anchor\ Weight = \begin{cases} 10 & case A \\ 5 & case B\ and\ case C \\ 1 & other \end{cases} \quad (2)$$

CaseA means that anchor is the entrance or exit,which is absolutely certain. CaseB indicates that only one anchor is retrieved for the operating vehicle, and the relationship between CBSID and anchor is relatively determined at this time. CaseC indicates that the anchor is a highway intersection, which is also critical for trajectory reconstruction, so it is given a medium weight.

4) Extracting complete driving route between high-weighted anchors from the road network: The sequence of CBSIDs can be fairly sparse due to a low sampling rate. As a consequence, some anchors cannot be detected. For example, in Fig. 10, only CBSIDs 1 and 2 are collected. The fingerprint matching module only outputs anchors A and C, and the path generated is A→C, whereas the actual driving route is A→B→C or A→D→C. The algorithm cannot distinguish such paths due to the limited hints. In this case, the shortest path is calculated between the two anchors using the Dijkstra algorithm [55] based on the road network. The principle is summarized as Algorithm 2. In this way, the complete trajectory of the driving route is obtained: {A1, A2, A6, A7, A8, A9}.
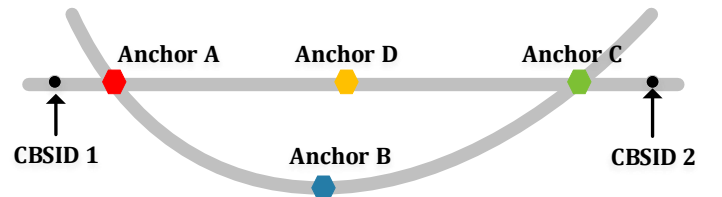

Fig. 10. Example of concatenating sequences of toll stations.

Algorithm 2: Complete driving route generation

Input: Road Network;
         the sequence of the concatenated
         anchors:$A = (A_1, A_2, \ldots, A_n)$.

Output: detailed driving route.

1 **for** $i \leftarrow 1$ **to** n **do**
2   **if** A_i *and* A_{i+1} *are not on the same road* **then**
3   search and add anchors between A_i and A_{i+1};
4   **else**
5   Dijkstra(A_i,A_{i+1});
         Search and add anchors between A_i and A_{i+1}
         by the shortest path.
6   **end**
7 **end**

IV. CASE STUDY

The study area is Hubei province, China. Its capital city, Wuhan, is known as the "thoroughfare leading to nine provinces". The highway network inside Hubei is also well developed.

A. Offline War-Driving

We traversed all major highways in Hubei for CBS fingerprints collection to construct the CBSID-anchor radiomap. During the war-driving, the car was driven at an average speed of 100 km/h. In the car, four mobile phones (MI-ONE C1, MI 2C, MI 1S, and SAMSUNG GT-i9100) were carried to record CBS signals and GNSS coordinates at an average interval of 5 s. GNSS coordinates, as explained in Section III, serve as the sampling points in fingerprint clustering. The anchors along highways were manually recorded. Following this procedure, 332,510 pieces of data samples were collected, containing 487 anchors and 12,714 CBSs in total. The distribution of highways is shown in Fig. 11. There are no anchors on some highways because these highways were under construction in that period.

B. Parameter Setting

The CBS coverage is the coverage of each base station, expressed as the longest distance L that was continuously collected at the time of sampling (the vehicle constructing the offline map will lose track of the previous base station only when it travels more than L in the highway, i.e., it does not receive its signal). For example, in the Figure 6 for base station 04 (CBS 04), its maximum coverage is the distance from the sampling point P1 to P5 ($L_{P1} - L_{P5}$). The statistical results (Fig. 12) show that 96.3% of the base stations have a coverage range within 11,403m. Therefore, in Algorithm 1 we set the *threshold* parameter to 11,403, and when the distance between two toll stations is within 11,403m, we associate the base station (CBS) with both toll stations (anchor); otherwise, we associate the base station (CBS) only with the closer toll station (anchor), and not with the more distant one at the same time (as shown in Figure 6).

C. Radiomap Construction

The CBSID-anchor radiomap was constructed according to Algorithm 1. Fig. 12 depicts the result of radiomap construction. The black circles represent anchors, and the colored circles illustrate the pseudo-locations of CBSs, which are the average of all the GNSS coordinates of related sampling points. The CBSs with smaller LAC are painted in green, the larger in red. Within a similar color range, there are generally 1–3 anchors, which indicate that, by searching the LAC-anchor database first, the target anchors can be quickly locked.

D. Experiment and Evaluation

1) Evaluation metrics: The commonly used evaluation metrics, precision and recall, were adopted to evaluate the performance of the map matching algorithm [48], [49] . As shown in Fig. 13, given the ground truth route G, output route O, and the correctly matched part M, precision is defined as the ratio between M and O, and recall as the ratio between M and G.

2) Test datasets: Both simulated routes and field routes were tested in the experiment. The simulated routes were generated from data samples collected during war-driving. As described above, for each route during the war-driving, the operator recorded all the anchors along the route while collecting CBS signals every 5 second. Hence, the ground truths can be represented by the explicitly recorded sequences of anchors. CBS fingerprints were resampled at intervals of 30 second, 1 min, 5 min, 10 min, 15 min, and 20 min. In this way, the dataset of routes was obtained, consisting of 197 ground truth routes (sequences of anchors) and simulated sequences of CBSIDs (sampled at different time intervals). The field dataset includes three routes located in the central area of Wuhan where anchors are densely distributed. The ground truths are also made up of manually recorded sequences of anchors. Signal fingerprints are CBSIDs collected at an interval of 15 min.

3) Results: Fig. 14 shows the ratio of accurately recognized routes at different sampling time intervals. Apparently, the ratio slightly decreases as the sampling interval of CBSIDs increases. Within the interval of 5 min, the algorithm can accurately recover 99% of the 197 routes. It then takes a decreasing trend and drops to 93.4% at the sampling interval of 20 min.

There are thirteen mismatched routes for the dataset of the 20-minute sampling interval. And the mismatched parts of these routes cover the cases of route mismatching for datasets sampled at other intervals. The detailed information of the mismatched parts of routes are given in Fig. 15 and Table I.

TABLE I
MISMATCHED PART OF ROUTES

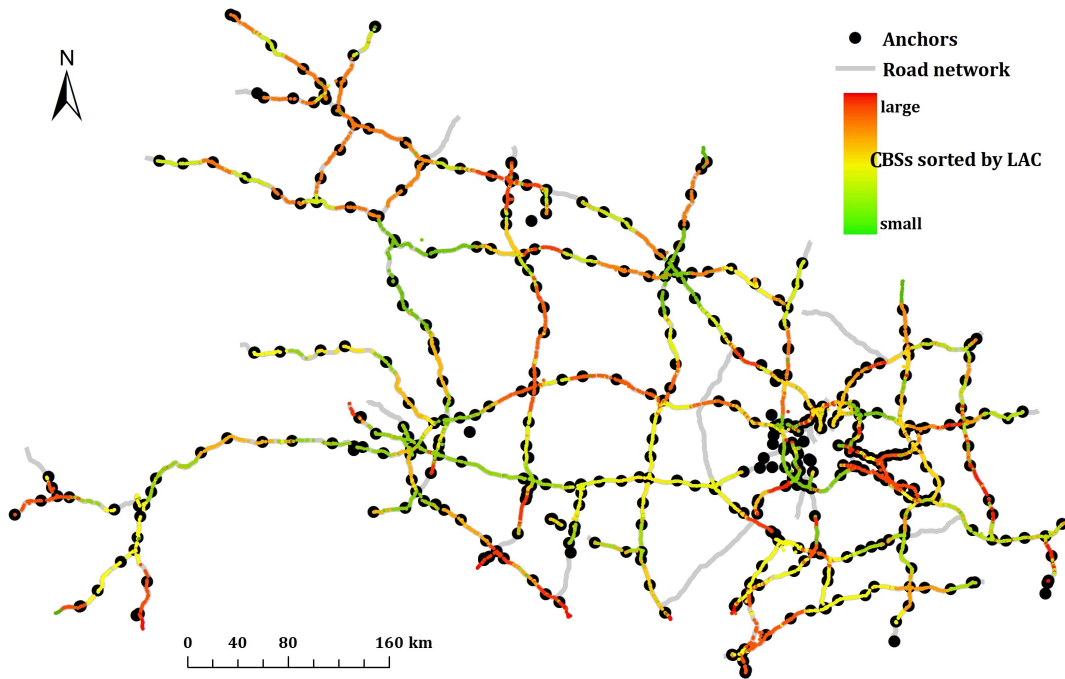| Area	Ground truth	Output route
A	367-170-169	367-368-169
B	361-L2-363-364-L1-228	361-L2-L1-228
C	129-L4-137-L5-128	129-L5-128
	136-L5-L3-L4-55	136-L5-137-L4-55
D	268-292-293-294-263-15-14	268-16-15-14
	264-15-263-294-293-292-268	264-15-16-268

Fig. 11. Distribution of anchors and pseudo-localization of CBSs colored by LAC.
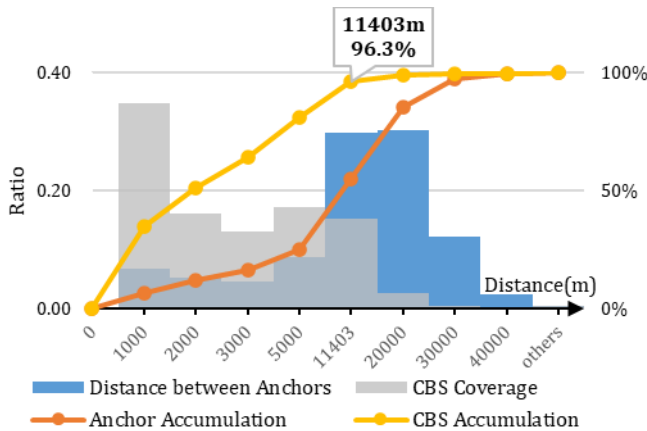

Fig. 12. Statistical result of the CBS coverage ranges on highways and the distances between anchors.
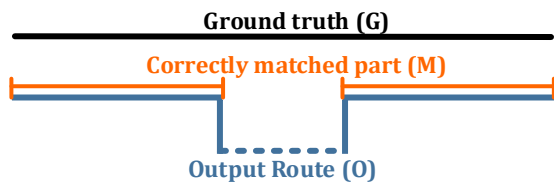

Fig. 13. Definition of precision M/O and recall M/G.

10 min, we get three more mismatched routes. One of them is located in area B, and the other two are distributed in area C. The algorithm tends to output the shorter path in these two areas. This is due to the fact that no CBSID is collected in the area because of the sparse sampling interval, and the algorithm has to calculate the shortest path. The same mismatches take place in area C as the sampling interval rises to 15 min. Area D is larger than the above-discussed three areas. When the sampling interval reaches 20 min, mismatches begin to take place in this area.
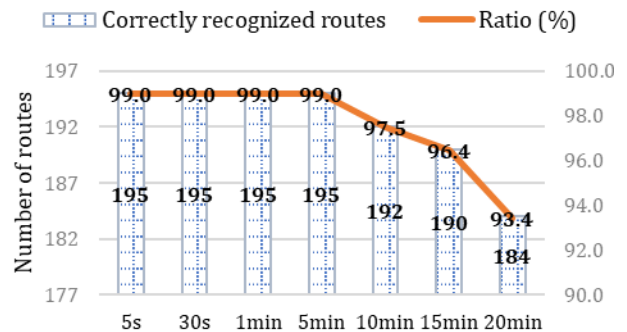

Fig. 14. Testing result of simulated routes. Horizontal axis represents sampling time interval.

The distribution of three field routes is shown in Fig. 16. Route 1 starts and ends at toll station 16, taking a U-turn at toll station 5. A repeat trial was conducted to evaluate the robustness for each route. In total, six routes were examined. All routes consist of repeated sections generated by deliberately taking U-turns. Experiment results are presented in Table II. The formula for the criterion (P,R) to measure the quality of

For all sampling intervals, two routes were not accurately recognized, the mismatched parts of which are the same and distributed in area A. The ground truth route is 367-170-169, whereas the output route is 367-368-169. These two routes are less than 1 km apart. When the sampling interval increases to

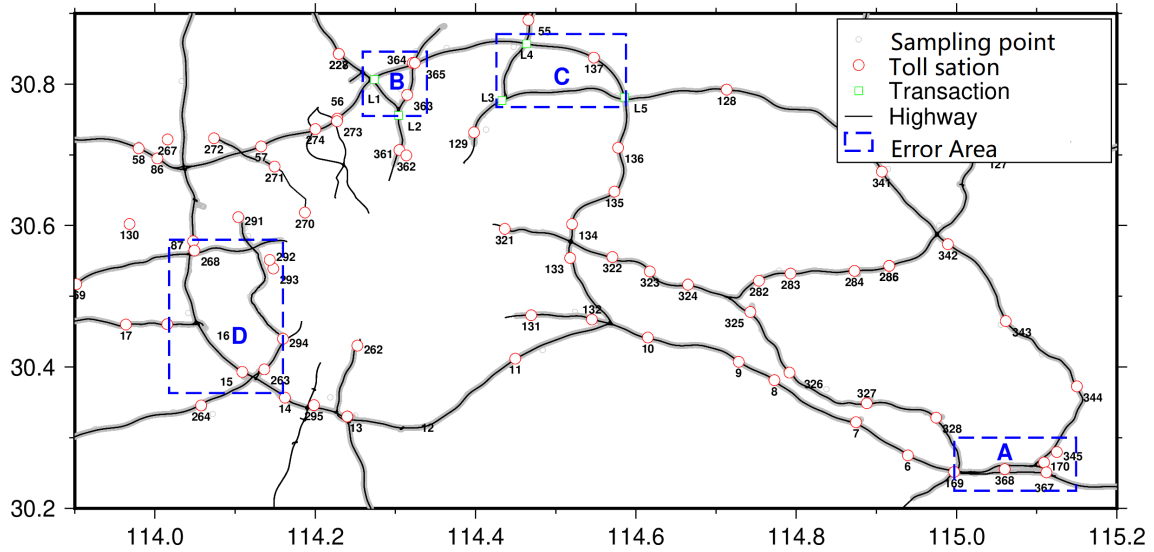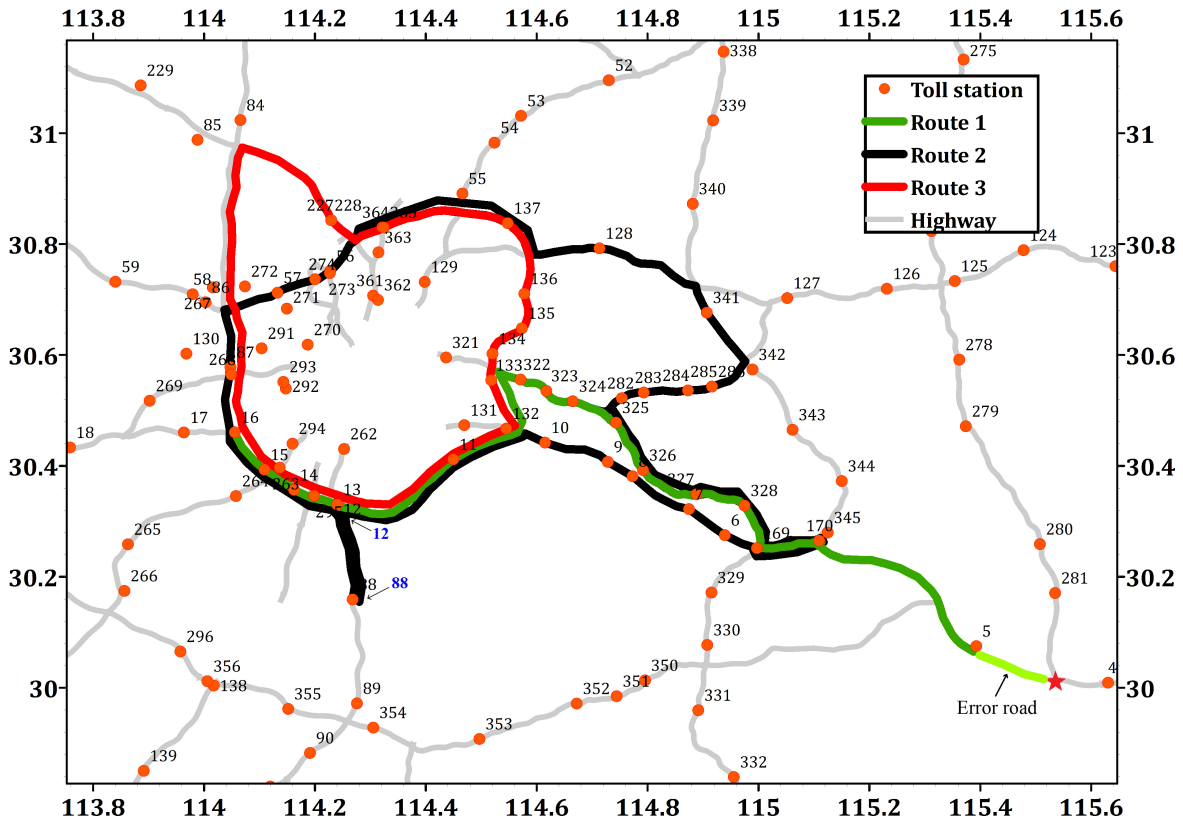Fig. 15. Distribution of the error areas.

Fig. 16. Distribution of field routes (In order to test the robustness of the algorithm and its ability to recognize off-ramps, we conducted tests in which the vehicles were driven on special trajectories whenever possible. Our method can accurately reconstruct most of the driving trajectories, but there is also an error road as marked in the lower right corner).

the calculation is shown in Equation 3 and Equation 4.

$$Precision = \frac{True\ Positive}{True\ Positive + Fasle\ Positive} \quad (3)$$

$$Recall = \frac{True\ Positive}{True\ Positive + Fasle\ Negative} \quad (4)$$

TABLE II
TESTING RESULTS OF FIELD ROUTES

Route	Length (km)			Number of CBSIDs		Result	
	G	O	M	Received	Valid	P	R
1	379.8	410.6	379.8	29	25	0.92	1
	379.8	410.6	379.8	29	25	0.92	1
2	370.4	370.4	370.4	31	26	1	1
	370.4	370.4	370.4	32	28	1	1
3	249.1	249.1	249.1	15	13	1	1
	249.1	249.1	249.1	15	12	1	1

As shown in Table II, the number of valid CBSIDs is less than that of received CBSIDs, indicating that, in practice, some CBSIDs cannot find matches due to incomplete radiomap. With the limited number of available CBSIDs, routes 2 and 3 are recognized with an accuracy of 100%. The average precision is 0.97. Routes were deliberately twisted by taking U-turns, such as the route between toll station 88 and 12, whereas the algorithm does not fall into the trap and still successfully recognizes all these small twists.

The mismatched part of route 1 is located between toll station 5 and the starred intersection (Fig. 16). This is due to the fact that the collected CBSID actually covers these two anchors, and the intersection is highly weighted in the trajectory mapping process. In practice, drivers are not likely take repeated routes like Route 1, and thus such a mistake is rare.

We compare our algorithm with the traditional Dijkstra algorithm and list the advantages of our algorithm in the following two common cases (as shown in Figure 17). In the CaseA, Dijkstra algorithm incorrectly divides the benefits that originally belong to Highway A to the investors of Highway B. In the CaseB, the Dijkstra algorithm fails to recognize the U-shaped trajectory of the vehicle, resulting in a reduction in the revenue that would otherwise belong to the highway investor.

Our method could effectively solve the above two common cases, while the traditional Dijkstra algorithm does not accurately restore the trajectory of vehicle operation, even in the worst case (no base station signal is received and two highways are densely adjacent), our method is no worse than the traditional Dijkstra method.

V. CONCLUSIONS

In this paper, a CBS fingerprint-based method was proposed for vehicle route recognition on the highway. The method generates a vehicle's driving route by 1) periodically collecting the CBSIDs of register CBSs along the highway; 2) converting the sequence of CBSIDs to the real route through fingerprint matching and trajectory mapping. Experiments

were conducted on both simulated and field routes. In the experiment on 197 simulated routes, CBSIDs were collected at the intervals of 5 s, 0.5 min, 1 min, 5 min, 10 min, 15 min, and 20 min, and the percentages of accurately matched map routes were 99%, 99%, 99%, 99%, 97.5%, 96.4%, and 93.4%, respectively, showing strong robustness against temporal and spatial sparseness of fingerprints. In the experiment on three field routes, CBSIDs were collected every 15 min, and routes were deliberately twisted by taking U-turns. The algorithm still accurately recognized two of the routes, and the precision and recall of the other route were 0.92 and 1, respectively. Experimental results show that the algorithm can accurately recover the vehicle's route from a sparse and noisy CBS fingerprint sequence.

The strengths of this present work are as follows.

After carefully examining the features of available localization methods, i.e., GNSS, RFID, LPR, and vision, motion, and signal fingerprints, we concluded that the proposed CBS fingerprint-based method is most appropriate for tracking a vehicle's route on the highway, in consideration of its low energy consumption, free infrastructure installation, less latency time, and general availability. The method has been adopted by the Transportation Bureau of Hubei Province and has been applied in the local toll collection system.

The proposed method is simple and efficient. It requires only the CBSIDs of register CBSs, which permits its wider range of applications in digital devices since the availability of CBS information is sometimes limited. Vehicles' special mobility patterns on the highway are also used in developing the route recognition method. The method spares the effort of converting fingerprints into coordinates and improves the fingerprint matching accuracy by the incorporation of spatial constraints.

There are also some limitations to this present work. One such limitation is that the route recognition method cannot distinguish multiple paths in small areas like the case shown in Fig. 15. In such cases, the shortest path that the driver may prefer in practice is computed. Further testing of the algorithm on smaller areas and addressing the issue are being considered.

The other limitation concerns the radiomap construction. The offline war-driving is time-consuming, and it is not easy to update the constructed radiomap in real time (The study area includes 332,510 pieces of data samples, 487 anchors and 12,714 CBSs in total). Therefore, building a more effective radiomap needs to be suggested further. Electronic cards are currently being distributed to drivers to only collect CBSIDs. Later, transplanting it onto smartphones to collect RSS as well as other auxiliary information is being considered, encouraging users to help populate the radiomap in real time [56].

REFERENCES

[1] T. L. McDaniel, "The (r) evolution of toll-collection technology," *IEEE Potentials*, vol. 34, no. 5, pp. 34–39, 2015.
[2] S. Naaz, S. Parveen, and J. Ahmed, "An artificial intelligence based toll collection system," *ICIDSSD 2020*, p. 389, 2021.
[3] M. N. H. Mir, N. I. Kayesh, T. M. Khan, and A. Sattar, "Iot based digital toll collection system: A perspective," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, 2021, pp. 1406–1410.
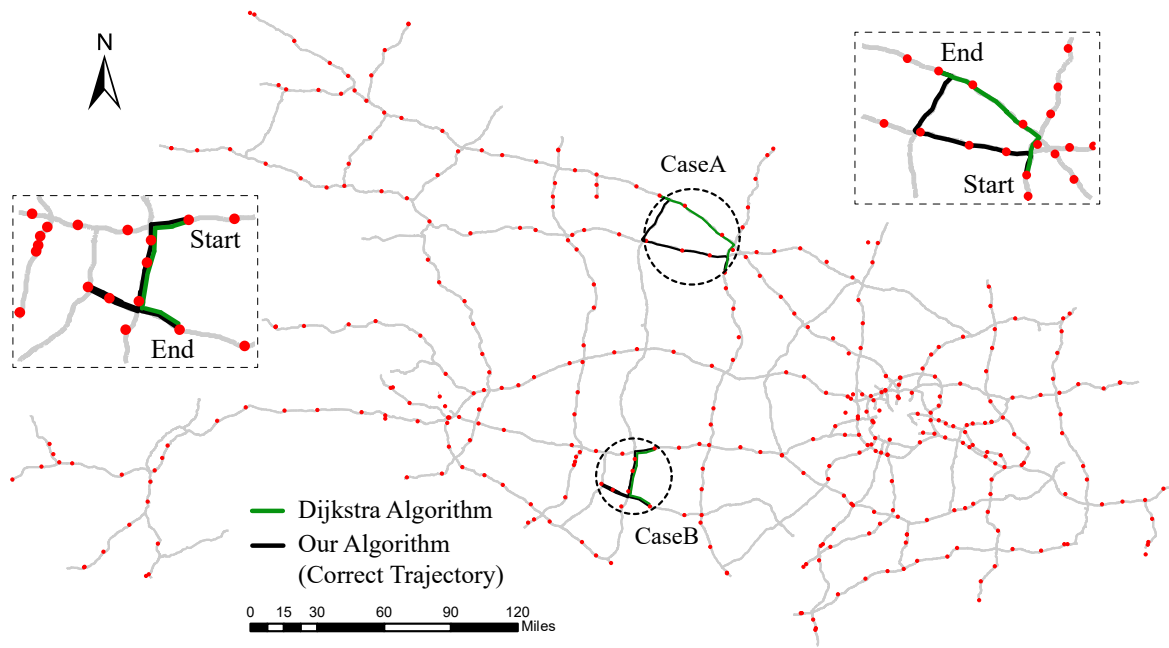
Fig. 17. Comparison of trajectory reconstruction based on CBSID sequence using Dijkstra algorithm and our algorithm.

[4] Z. Liu, M. Sang, and L. Wu, "Developing history and trends of chinese expressway's toll collection techniques," in *2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2017, pp. 219–222.

[5] M. Sun, W. Wang, P. Zheng, D. Luo, and Z. Zhang, "A novel road energy harvesting system based on a spatial double v-shaped mechanism for near-zero-energy toll stations on expressways," *Sensors and Actuators A: Physical*, vol. 323, p. 112648, 2021.

[6] M. Handte, S. Foell, S. Wagner, G. Kortuem, and P. J. Marrón, "An internet-of-things enabled connected navigation system for urban bus riders," *IEEE internet of things journal*, vol. 3, no. 5, pp. 735–744, 2016.

[7] W. Tu, F. Xiao, L. Li, and L. Fu, "Estimating traffic flow states with smart phone sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 126, p. 103062, 2021.

[8] A. Viel, D. Gubiani, P. Gallo, A. Montanari, A. Dalla Torre, F. Pittino, and C. Marshall, "Map matching with sparse cellular fingerprint observations," in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*. IEEE, 2018, pp. 1–10.

[9] M. Castro, L. Iglesias, J. A. Sánchez, and L. Ambrosio, "Sight distance analysis of highways using gis tools," *Transportation research part C: emerging technologies*, vol. 19, no. 6, pp. 997–1005, 2011.

[10] F. Zhou, L. Feng, P. Yu, W. Li, X. Que, and L. Meng, "Drl-based low-latency content delivery for 6g massive vehicular iot," *IEEE Internet of Things Journal*, 2021.

[11] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3607–3644, 2018.

[12] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson, "Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, 2011, pp. 68–81.

[13] G. Ananthanarayanan, M. Haridasan, I. Mohomed, D. Terry, and C. A. Thekkath, "Startrack: a framework for enabling track-based applications," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, 2009, pp. 207–220.

[14] Y. Man and E. C.-H. Ngai, "Energy-efficient automatic location-triggered applications on smartphones," *Computer Communications*, vol. 50, pp. 29–40, 2014.

[15] A. Tahat, G. Kaddoum, S. Yousefi, S. Valaee, and F. Gagnon, "A look at the recent wireless positioning techniques with a focus on algorithms for moving receivers," *IEEE Access*, vol. 4, pp. 6652–6680, 2016.

[16] D. Kim, S. Lee, and H. Bahn, "An adaptive location detection scheme for energy-efficiency of smartphones," *Pervasive and Mobile Computing*, vol. 31, pp. 67–78, 2016.

[17] W. Liu, H. Yang, and Y. Yin, "Efficiency of a highway use reservation system for morning commute," *Transportation Research Part C: Emerging Technologies*, vol. 56, pp. 293–308, 2015.

[18] S. Spinsante and C. Stallo, "Hybridized-gnss approaches to train positioning: challenges and open issues on uncertainty," *Sensors*, vol. 20, no. 7, p. 1885, 2020.

[19] M. Deriche and M. Mohandes, "A hybrid rfid-lpr system for vehicle access control during pilgrimage season in saudi arabia," in *International Multi-Conference on Systems, Signals & Devices*. IEEE, 2012, pp. 1–6.

[20] A. Roy, J. Siddiquee, A. Datta, P. Poddar, G. Ganguly, and A. Bhattacharjee, "Smart traffic & parking management using iot," in *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2016, pp. 1–3.

[21] L. Xu, W. Zhuang, G. Yin, D. Pi, J. Liang, Y. Liu, Y. Lu *et al.*, "Geometry-based cooperative localization for connected vehicle subject to temporary loss of gnss signals," *IEEE Sensors Journal*, vol. 21, no. 20, pp. 23 527–23 536, 2021.

[22] W. Zhang, B. Lin, C. Gao, Q. Yan, S. Li, and W. Li, "Optimal placement in rfid-integrated vanets for intelligent transportation system," in *2018 IEEE International Conference on RFID Technology & Application (RFID-TA)*. IEEE, 2018, pp. 1–6.

[23] A. H. Ashtari, M. J. Nordin, and M. Fathy, "An iranian license plate recognition system based on color features," *IEEE transactions on intelligent transportation systems*, vol. 15, no. 4, pp. 1690–1705, 2014.

[24] Q. D. Vo and P. De, "A survey of fingerprint-based outdoor localization," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 491–506, 2015.

[25] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach, "Mobile visual location recognition," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 77–89, 2011.

[26] H. Wang, D. Zhao, H. Ma, and Y. Liang, "Common crucial feature for crowdsourcing based mobile visual location recognition," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 908–912.

[27] T. Guan, Y. He, L. Duan, J. Yang, J. Gao, and J. Yu, "Efficient bof generation and compression for on-device mobile visual location recognition," *IEEE MultiMedia*, vol. 21, no. 2, pp. 32–41, 2013.

[28] Y. Zhang, N. Chen, W. Du, Y. Li, and X. Zheng, "Multi-source sensor based urban habitat and resident health sensing: A case study of wuhan, china," *Building and Environment*, vol. 198, p. 107883, 2021.

[29] C. Bo, T. Jung, X. Mao, X.-Y. Li, and Y. Wang, "Smartloc: Sensing landmarks silently for smartphone-based metropolitan localization,"

EURASIP Journal on Wireless Communications and Networking, vol. 2016, no. 1, pp. 1–17, 2016.

[30] H. Aly and M. Youssef, "Dejavu: an accurate energy-efficient outdoor localization system," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2013, pp. 154–163.

[31] 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 *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2733–2741.

[32] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2016.

[33] C.-W. Ang, "Vehicle positioning using wifi fingerprinting in urban environment," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. IEEE, 2018, pp. 652–657.

[34] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," 2011.

[35] A. Anjomshoaa, F. Duarte, D. Rennings, T. J. Matarazzo, P. deSouza, and C. Ratti, "City scanner: Building and scheduling a mobile sensing platform for smart city services," *IEEE Internet of things Journal*, vol. 5, no. 6, pp. 4567–4579, 2018.

[36] Q. Li, L. Zhang, and X. Wang, "Loosely coupled gnss/ins integration based on factor graph and aided by arima model," *IEEE Sensors Journal*, vol. 21, no. 21, pp. 24 379–24 387, 2021.

[37] M. Ibrahim and M. Youssef, "Cellsense: An accurate energy-efficient gsm positioning system," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 286–296, 2011.

[38] M. Chaturvedi and S. Srivastava, "Real time vehicular traffic estimation using cellular infrastructure," in *2013 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2013, pp. 1–6.

[39] T. Wigren, "Adaptive enhanced cell-id fingerprinting localization by clustering of precise position measurements," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 3199–3209, 2007.

[40] M. Bshara, U. Orguner, F. Gustafsson, and L. Van Biesen, "Fingerprinting localization in wireless networks based on received-signal-strength measurements: A case study on wimax networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 283–294, 2009.

[41] S. C. Ergen, H. S. Tetikol, M. Kontik, R. Sevlian, R. Rajagopal, and P. Varaiya, "Rssi-fingerprinting-based mobile phone localization with route constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 423–428, 2013.

[42] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*, vol. 2. Ieee, 2000, pp. 775–784.

[43] M. A. Youssef, A. Agrawala, and A. U. Shankar, "Wlan location determination via clustering and probability distributions," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003)*. IEEE, 2003, pp. 143–150.

[44] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-id sequence matching," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011, pp. 293–306.

[45] P. Zhou, Y. Zheng, and M. Li, "How long to wait? predicting bus arrival time with mobile phone based participatory sensing," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 2012, pp. 379–392.

[46] G. Liu, J. Liu, F. Li, X. Ma, Y. Chen, and H. Liu, "Subtrack: Enabling real-time tracking of subway riding on mobile devices," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2017, pp. 90–98.

[47] T. F. Smith, M. S. Waterman *et al.*, "Identification of common molecular subsequences," *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.

[48] R. Mohamed, H. Aly, and M. Youssef, "Accurate real-time map matching for challenging environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 847–857, 2016.

[49] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 336–343.

[50] A. Dalla Torre, P. Gallo, D. Gubiani, C. Marshall, A. Montanari, F. Pittino, and A. Viel, "A map-matching algorithm dealing with sparse cellular fingerprint observations," *Geo-spatial Information Science*, vol. 22, no. 2, pp. 89–106, 2019.

[51] G. R. Jagadeesh and T. Srikanthan, "Online map-matching of noisy and sparse location data with hidden markov and route choice models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2423–2434, 2017.

[52] G. Schulze, C. Horn, and R. Kern, "Map-matching cell phone trajectories of low spatial and temporal accuracy," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 2707–2714.

[53] Z. Xiao, F. Li, R. Wu, H. Jiang, Y. Hu, J. Ren, C. Cai, and A. Iyengar, "Trajdata: On vehicle trajectory collection with commodity plug-and-play obu devices," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9066–9079, 2020.

[54] Y. Zhang, X. Zheng, M. Chen, Y. Li, Y. Yan, and P. Wang, "Urban fine-grained spatial structure detection based on a new traffic flow interaction analysis framework," *ISPRS International Journal of Geo-Information*, vol. 10, no. 4, p. 227, 2021.

[55] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[56] N. Yu, Y. Li, X. Ma, Y. Wu, and R. Feng, "Comparison of pedestrian tracking methods based on foot-and waist-mounted inertial sensors and handheld smartphones," *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8160–8173, 2019.

ACKNOWLEDGMENT

YINGBING LI was born in Hubei, China, in 1972. He received the Ph.D. degree in geodesy and geomatics from Wuhan University, China, in 2003. Since 2001, he has been teaching with Wuhan University, and in 2008 and 2013, he visited Ohio State University twice as a Visiting Scholar. His main research field are typical natural disaster evolution process analysis, emergency response scenario deduction and spatial big data analysis.

YAN ZHANG (ORCID: 0000-0002-2059-4171) received the B.S. and M.S. degrees in Geographic Information System engineering from the Wuhan University. He now studies at State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing and is a visiting scholar at the National University of Singapore now. His research interests are natural language processing, complex network and spatial data analysis.

MIN CHEN (ORCID: 0000-0003-0565-8322) is a graduate student in School of Geodesy and Geometrics, Wuhan University. She is major in Cartography and Geographic Information Science, and her main research field is cascading disaster modelling and analysis from the geographic perspective.