Comparative Study and Application-Oriented Classification of Vehicular Map-Matching Methods

Matěj Kubička,
Hugues Mounier, and Silviu-Iulian Niculescu
Laboratoire des Signaux et Systemes, CNRS/CentraleSupélec, 91192, Gif-Sur-Yvette, France. E-mail: matej.kubicka@lss.supelec.fr, hugues.mounier@lss.supelec.fr, silviu.niculescu@lss.supelec.fr

Arben Cela
UPE, ESIEE Paris, 93162, Noisy-Le-Grand, France, E-mail: arben.cela@esiee.fr

IMAGE LICENSED BY INGRAM PUBLISHING

***Abstract*—Vehicular map-matching aims to identify the position of a vehicle in a road-network. Different map-matching applications make use of different formulations of the problem leading to some ambiguous usage. This work intends to make a step towards resolution of this issue. We cross-link map-matching methods, classes and applications in order to provide insight into this area of study. A selection of current map-matching methods is reviewed, classified and appropriately linked with other methods. Conclusions on trends and influential ideas are drawn with respect to the specific flavor of individual applications. Finally, the trade-offs that must be considered when selecting a map-matching method are discussed and selection guidelines are provided.

Digital Object Identifier 10.1109/MITS.2018.2806630
Date of publication: 20 April 2018

1939-1390/18©2018IEEE
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

I. Introduction

Roughly speaking, map-matching aims to identify vehicle position or route on a map [4], [48], [66]. Figure 1 shows its role in the context of navigation systems and other services. Map-matching has been under active research since the dawn of the global navigation satellite systems in 1990's. A few contributions dating before that (e.g. [25]) exist, but the interest in the context of vehicular navigation appeared after GPS became widely available. Original interest was in connection with navigation assistants, but a large area of applications emerged after the mobile internet became available. One can cite, among others, location-based services, floating car data, "pay-as-you-go" services, automatic emergency beacons and fleet surveying. An exhaustive list of map-matching applications was given by Velaga in his thesis [57].

When the map and the positioning data are sufficiently accurate, the problem is trivial. Hence, it might seem that map-matching is not a difficult problem. Nowadays, we have satellite navigation with sub-meter precision and detailed maps on scales allowing to distinguish millimeters. However, neither of the preceding is guaranteed and errors in both can be intricate. The satellite-based positioning systems are unreliable, especially in urbanized areas where satellite visibility is limited and reflected satellite signals are common. This leads to outliers in positioning data and to missing samples when the satellite signal is weak. This in itself can make correct matching difficult in some situations. However, the map can also be outdated (missing some roads), or the map model itself can introduce ambiguity into the matching process. Commonly used maps are directed graphs where junctions are represented by their center points and roads are represented by polygonal curves. Such a simplistic representation of otherwise complex road network can make correct matching difficult. Consequently, map-matching is not always trivial. Over hundred of different methods have been published in the last two decades. They are based on various methods and approaches: from simple geometry and topology to advanced concepts that involve fuzzy logic, particle filtering, belief theory, conditional random fields, Kalman filtering, neural networks, genetic algorithms, ant colony optimization and others.

By inspecting the results concerning the map-matching problem, it is not immediately clear how the performance should be evaluated. This fact leads to different evaluation methodologies proposed by different authors. Unfortunately, these are often not mutually compatible. For example, some authors evaluate map-matching performance as a ratio of correctly matched positions with respect to the total number of samples while other authors consider the degree of overlap between the matched route and the correct route. There is no common basis between the two methodologies and hence direct comparison seems not possible. This makes it difficult to argue on which method is best suited for any given use.

To the best of the authors' knowledge, there exist two reviews on map-matching: the first one was published in 2007 [48], the second one in 2014 [24]. There were significant advances in the field since the first review. The second review, on the other hand, is mainly concerned with online map-matching. This paper reviews map-matching from an application-oriented perspective that was not taken into account in the previous reviews. Three generic application types are identified (navigation, tracking and mapping), discussed and then matched to different map-matching classes. Then a selection of methods is reviewed, with emphasis on novel ideas and promising research directions. Finally, the trade-offs that have to be considered when selecting a map-matching method are discussed and selection guidelines are provided. We hope that both researchers and engineers will find it useful: researchers when they aim to design a new method specific for their needs and engineers when looking for the most appropriate method for their use case.

The text is organized as follows. Related work is reviewed in Section II. The section III discusses common map-matching applications and clusters them based on commonalities they share. The section IV presents used terminology. The section V presents a classification based on dictionary used in published articles. The section VI discusses standing research challenges with emphasis on the problems that are either overlooked or difficult to deal with. It reviews a number of map-matching methods with a focus on the research challenges each method tries to solve. A selection of published map-matching methods is reviewed in Section VII. Some of the methods that were already discussed in the Section VI are returned to and reviewed from application-oriented point of view. Guidelines for method selection and discussion of related trade-offs is in Section VIII. The summary is in Section IX.

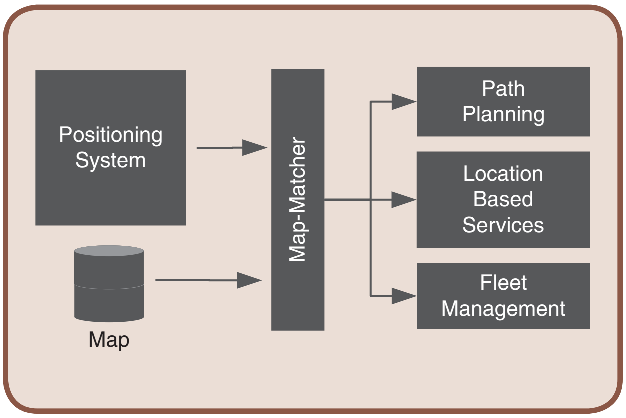

FIG 1 Map-matching converts raw position to position on the map (originally printed in [35]).

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

II. Related Work & Framework

As mentioned in the introduction, Quddus et al. [48] published an overview of map-matching methods in 2007. The authors classify them as *geometric*, *probabilistic*, *topological* and *advanced*. More precisely, geometric methods make use of geometric closeness criteria for matching, topological methods leverage contiguity of arcs in road-network graphs and probabilistic methods make use of uncertainties in reported position to approximate error-bounding regions. Advanced methods group together other methods that do not fit this classification scheme otherwise, as for example methods based on fuzzy logic or on Kalman filtering. Most of the methods published in the last fifteen years could be viewed as advanced since recent map-matching algorithms often combine the ideas of geometric, topological and sometimes probabilistic methods.

The most recent review of map-matching methods was proposed by Hashemi and Karimi [24] in 2014. The authors focused on map-matching for navigational applications. They reviewed a selection of publications and categorized them as *simple*, *weight-based* and *advanced*. The motivation for this categorization and used classification criteria is not clearly explained to the reader.

Wei et al. provide a short review of map-matching methods in [62]. The authors classify them as *incremental max-weight*, *global max-weight* and *global geometric*. This was adopted by other authors since (e.g. [37]). The max-weight methods search for the route that maximizes some fitness function. The geometric methods search for a route geometrically similar to given trajectory. The incremental methods match individual samples while still collecting them. The global methods process complete trajectory after it was collected.

Quddus et al. list performance of ten map-matching methods in [48]. Hashemi and Karimi list performance of nineteen methods in [24]. The performance figures listed in these two reviews are based on results reported in the reviewed papers. Wei et al. present their comparison of fifteen methods in [63]. The authors implemented their own versions and tested them using unified methodology which assures comparability of the results.

This review differs with respect to the cited reviews with its applications-oriented point of view. The authors of published methods use varying and often incompatible formulations of the map-matching problem. These differences are conveniently explained by specific needs of different applications. Hence we first review map-matching classes and applications. This allows us to expose the problem in its full size. A large portion of the text is devised to linking, cross-linking and highlighting relationships that exist between different methods, classes and applications. This allows us to draw conclusions with respect to each application and to discuss the trade-offs that must be considered when selecting a method for particular use.

The review is restricted to outdoor vehicular methods. While there is a number of works on indoor map-matching, it has little in common with outdoor map-matching. The subject of simultaneous localization and mapping (SLAM) is not in scope of this review. While SLAM is related to map-matching, there is little overlap in literature. A number of other published methods is not covered in this review as they didn't fit into our exposition. Most notably these are methods based on fuzzy logic [21], [29], [31], [46], [47], [54], low-sampling rate methods [13], [38], [50], [69], [71], methods based on particle filters [8], [9], [16], methods based on belief theory [19], [40], [41], [68] and methods that use a probabilistic map-matching framework pioneered by Bierlaire et al. [5], [6], [14]. Most of these methods were already covered in previous reviews.

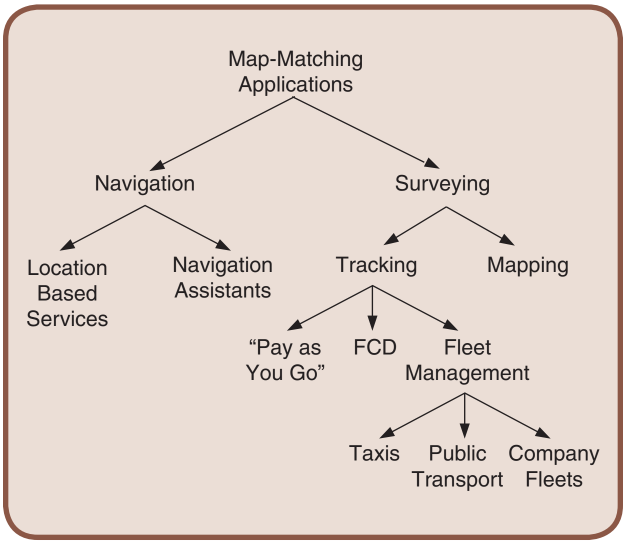

FIG 2 Relations between map-matching applications.

III. Applications

Vehicular applications are either for *navigation* or for *surveying*. Surveying applications further divide between *tracking* and *mapping*. A list of applications bundled according to these application areas is given in Figure 2.

A. Map-Matching for Navigation

Navigational applications make use of real-time positioning data to provide guidance or location-relevant information. Navigation assistants belong to this class. Most other navigational applications belong to the group of *location based services* (LBS). Their definition according to Virrantaus [59] reads "LBSs are information services accessible with mobile devices through the mobile network and utilizing the ability to make use of the location of the mobile device". Hence, they provide answers to questions like "Where am I?", "Where are my friends?", "What is here around me?" [53].

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

Methods for navigational applications often view map-matching as matching of the current position only. This is in contrast with tracking applications where the goal is to recover vehicle route. The typical trajectory sampling rate is once per second. Again, this is in contrast with tracking applications where sampling periods longer than thirty seconds are common. Many navigational methods also require supplementary information such as uncertainty in position, speed and dilution of precision coefficients. These are often used as aids for matching individual samples.

The satellite-based positioning systems are unreliable, especially in urbanized areas where satellite visibility is limited and reflected satellite signals are common. This leads to outliers in positioning data and to missing samples when the satellite signal is weak.

Some authors view map-matching as a method to enhance accuracy of the satellite navigation system by restricting the vehicle to positions on roads in the map. This was largely abandoned since May 2000 when selective availability[1] was disabled. The accuracy of civilian GPS increased by two orders since (from roughly 100 meters to 1 meter most of the time [17]). However, there are exceptions. Map-matching methods based on Wi-Fi and Cell positioning still leverage the "being on the road" assumption to enhance accuracy [30], [55], [61]. Development of these methods is motivated by a short battery life of handheld devices due to a high energy consumption of GNSS receivers.

B. Map-Matching for Tracking

Map-matching for tracking aims to recover vehicle route from a sequence of sampled positions. This is in contrast with navigational applications which "snap" individual trajectory samples to the road-network. Floating car data systems use the route together with observed speed to estimate traffic state. The "pay as you go" services use it to collect toll fees. The fleet management uses it to track company vehicles and to generate itineraries. Taxi and public city transport companies use it to optimize their services.

Tracking methods typically collect trajectories with minimum detail. The trajectory is usually a sequence of observed positions (latitude, longitude) sampled with a period longer than 30 seconds. No additional information is collected to minimize transmission payload.

C. Map-Matching for Mapping

Map-matching is also used when integrating GNSS data into existing geographic information systems (GIS). It is typically used to locate new roads and to enter them in the GIS database. For a survey of these practices see National Cooperative Highway Research Program (NCHRP) Synthesis 301 on collecting, processing, and integrating GPS data into GIS [7].

Mapping tasks require positioning system that is able to produce an uninterrupted stream of position samples. This is different from navigational and most surveying applications, as they can often tolerate low sampling rate or loss of satellite signal. High sampling rate map-matching methods designed specifically to be robust against positioning errors are a good fit for mapping applications.

IV. Terminology

Let $G = (V, E)$ be a road-network graph where V is a set of nodes and E is a set of arcs. Each node has assigned position on Earth (denoted $\mathrm{loc}(v_k)$ for k-th node in V). Arcs represent roads between locations of the nodes the arcs connect. The road-network graph G is sometimes referred to as a "map" in the text. Similarly, arcs are occasionally referred to as "roads" to help understandability.

Let ordered set $s = (s_1, s_2, s_3, \ldots, s_n)$ be a sequence of sampled states s_i at discrete time instants $i \in \{1 \ldots n\}$. We refer to s_i as i-th *sample* and to s as *trajectory*. With map-matching for tracking applications (Section III-B) the trajectory is usually a sequence of sampled positions. In map-matching for navigation (Section III-A) each sample is typicaly a set that includes position information accompanied with additional information such as instantaneous speed and error bounds.

The navigational methods return matched position for each sample. Let such output be denoted as *matching* m and defined $m = (p_1, p_2, \ldots, p_n)$ where $p_i = (a_i, o_i)$ is i-th matched position on the road network. a_i is the road on which vehicle travels and o_i is the distance from upstream crossroad. The tracking methods do not match individual samples, instead they concentrate on travel route identification. Let such output be denoted *route* r and defined as a sequence of contiguous arcs $r = (e_1, e_2, \ldots)$ in G.

Finally, the path inference filter (Section VII-C) define output as a sequence of waypoints interleaved with paths between them. Let us denote such output as r' and define it $r' = (p_1, r_{12}, p_2, r_{23}, \ldots, p_n)$ where p_i's are the matched positions and r_{ij} is a sequence of contiguous arcs in E between p_i and p_j.

[1]Selective availability was artificially degrading positioning accuracy of civilian GPS to prevent its use for missile guidance.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

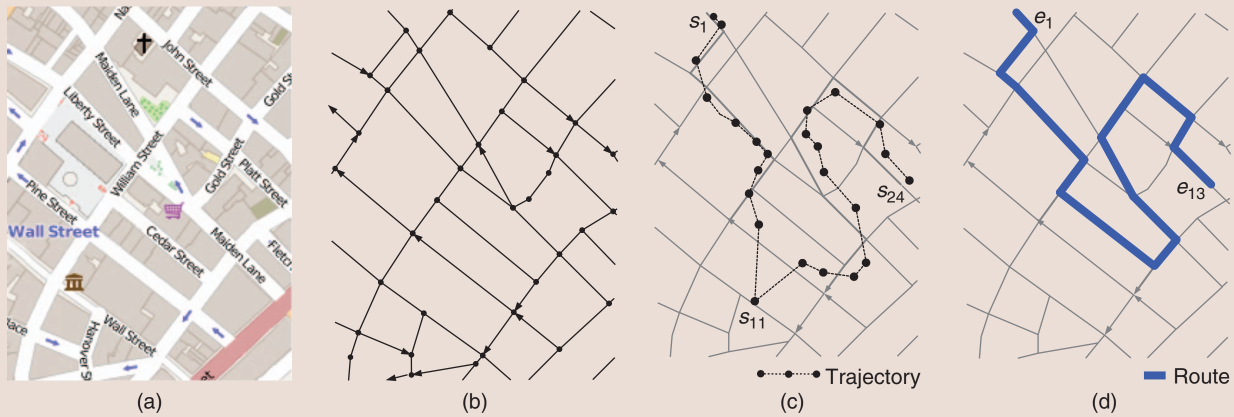

FIG 3 Example road-network, trajectory and map-matched route. (a) OpenStreetMap, (b) road-network $G = (V, E)$, (c) trajectory $s = (s_1, \ldots, s_{24})$, (d) route $r = (e_1, \ldots, e_{13})$.

For an example of a map, trajectory and route see Figure 3. Figure 3(a) shows OpenStreetMap render of a part of the Manhattan island in the New York, US (©OpenStreetMap contributors). Corresponding road network graph is given in Figure 3(b). Note the arrows, they indicate traffic direction on one-way streets. Missing arrow indicates a two-way street. Example trajectory made of samples $(s_1, s_2, \ldots, s_{24})$ is shown in Figure 3(c). A route matched to this trajectory is shown in Figure 3(d). Note that the sample s_{11} is an outlier: as can be observed in Figure 3(d) the route passes on Cedar Street rather on Wall Street, even the Wall Street is closer to sample s_{11} (compare to OpenStreetMap render in Figure 3(a)). This example is studied later in the text when effects of such outliers on some map-matching methods are discussed.

V. Classification

We introduce classification based on four criteria according to which most published methods describe themselves. These are later matched to different applications and matching techniques. See Table 1 for classification of map-matching methods with respect to their application.

- *Indoor/outdoor methods*—while outdoor map-matching makes use of satellite navigation [1], [42], [66], indoor map-matching has to use other positioning systems such as inertial navigation, radio beacons, etc. [67]
- *Pedestrian/vehicular/multimodal map-matching*—the pedestrian map-matching [51] is more challenging than the vehicular map-matching as pedestrians can be both indoors and outside. This is, strictly speaking, possible even with vehicular map-matching but authors often assume that vehicles are restricted to outdoor movement. The multimodal map-matching is used in the general case when the traveler can combine different travel modes [14] (such as traveling on a bicycle, in a vehicle or in public transportation).
- *Online/offline methods*—online map-matching [35], [66] methods output matched positions while still collecting the trajectory. Some authors use the term “real-time” instead of “online” [24], [35] to underline the fact that such map-matching returns results immediately with incoming samples. Other authors use the term “incremental” [37], [62]–[64]. Offline map-matching processes whole trajectory after it has been collected. Some authors use the term “global” instead of “offline” [37], [62]–[64].
- *Low/high sampling rate methods*—low sampling rate methods are commonly used for tracking. They are designed for situations where positioning data are sampled on periods longer than about thirty seconds (this number is somewhat arbitrary as authors consider various thresholds). There are methods designed specifically for low-sampling-rate applications [13], [38], [50], [69], [71] and also methods that try to be competitive over the full range of sampling rates [27], [42].

Table 1. Specifics of Different Applications.

	Navigation	Tracking	Mapping
returns	matched positions	route	both
domain	outdoor	outdoor	outdoor
sampling rate	high	low	high
use	online	offline	offline

VI. Research Challenges

A. Positioning Data and Maps

Errors in positioning data and in maps present the original research challenge in map-matching. GNSS navigation is

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

affected by environmental perturbations that cannot be corrected using augmentation systems (i.e. differential GPS). Even foliage can cause signal fading, and there can always be undetectable multipath and non-line-of-sight errors. These cannot be removed unless another positioning sensor is available. When it comes to maps, the classical raster maps were replaced by their vectorized counterparts. Each coordinate is stored using at least eight decimal places which enable sub-millimeter accuracy worldwide. Even we have tools to measure position highly accurately, practice shows that we cannot assume the maps are error-free. There is a problem of representation: in the most common case, the road network is reduced into a graph where each road is represented by a sequence of waypoints through which it passes. The road width, number of lanes, turn restrictions and other information is often omitted or incomplete. It can be argued that such information can be added. However the more details the map includes the more difficult it is to keep it up-to-date. It is also difficult to model map errors: published studies discuss the presence of random, systematic and modeling errors [15], [43], [70].

While positioning and map errors are difficult to observe individually, the incongruence between the trajectories and the routes can be observed. See Figure 4(b) for a histogram of *positioning errors* in dataset [34]. This dataset contains 2,695 kilometers of map-matched trajectories. These trajectories were sampled with various GPS and GLONASS receivers by volunteers worldwide [33]. The positioning error is the Euclidean distance between the trajectory sample and its matched position on the road-network. The histogram shows that 99% of the samples are less than sixteen meters away from matched position on the road-network. The histogram of *heading errors* with respect to vehicle speed is given in Figure 4(a). The heading error is the difference between the heading on individual samples and heading of the roads where these samples were matched to. The plot shows that heading information is not reliable at low speeds. This was previously observed by Quddus et al. [48].

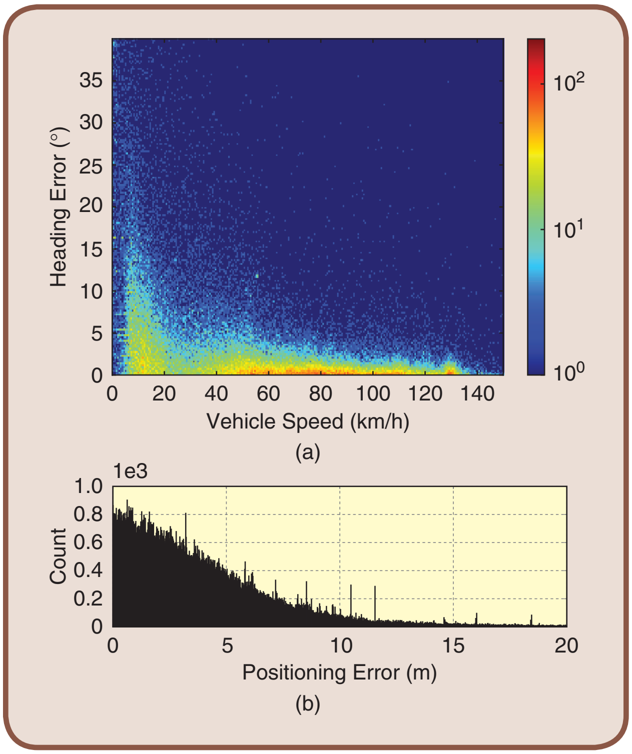

FIG 4 Plots of positioning and heading errors in published map-matching dataset [33]. (a) histogram of heading errors, (b) histogram of positioning errors.

B. Preprocessing

Preprocessing can be used to improve the performance of map-matching. It is relevant in the context of offline methods where a complete trajectory is available beforehand. NCHRP Synthesis 301 [7] studied the usage of map-matching methods by departments of transportation in the United States of America. According to their findings, the departments of transportation often preprocess the trajectory before doing the map-matching. Most methods published in literature ignore this step.

Newson and Krumm [42] argue that it is better to ignore some samples. Their preprocessing method moves through the trajectory sequence and removes samples within a radius of two standard deviations from the last accepted sample. The authors argue that there is low confidence that the apparent movement is due to actual vehicle movement rather than noise for the removed samples. As this reduces the number of samples in the trajectory the map-matching overhead is reduced as well.

Pink and Hummel [44] use extended Kalman filter with constant speed, constant yaw rate[2] vehicle model for trajectory preprocessing. It reads

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \\ \dot{v} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} -v\sin(\varphi) \\ v\cos(\varphi) \\ \eta \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

where x and y is position, φ is heading, η is yaw rate and v is vehicle speed. This smooths the trajectory because the Kalman filter combines the measured trajectory with the predictions of the model (which dictates constant speed and constant yaw rate). Note that this technique shall perform badly on sparsely sampled trajectories or when part of the trajectory is missing. This makes it suitable to mapping applications where the positioning system provides uninterrupted data stream.

C. Integrity Monitoring

Correct map-matching is not always possible when there is serious incongruence between the trajectory and the map

[2]vehicle turn rate.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

or when the situation is ambiguous. The reliability of map-matching in such situations is questionable. Hence, some map-matching methods perform *integrity monitoring* to report on the reliability of its output. This is often critical, especially in applications related to safety and electronic fee collection. The term integrity monitoring was imported from aerial navigation where integrity monitoring is used to verify the reliability of satellite navigation based position fixes. Integrity monitoring in context of map-matching can be formulated independently of the matching method. It takes the trajectory, the map and sometimes the map-matching output as input and provide *integrity indicators* that indicate how reliable the map-matching output is.

Integrity monitoring performance is typically evaluated with overall correct detection rate derived from counts of missed detections and false alarms. The results are sensitive to selected alarm levels against which the integrity indicators are compared. The sensitivity of the monitoring system is determined with them. They should be set experimentally such that both missed detections and false alarm ocurrences are kept at minimum. There is a trade-off: too many false alarms are due to high sensitivity while too many missed detections are due to low sensitivity. Note that in the context of map-matching there can always be unobservable missed detections since both positioning and map errors are not bounded. Seemingly congruent scenarios where the trajectory is aligned with the wrong roads can theoretically be observed. It implies that there can never be any kind of guarantee that the matching is correct.

The first mention of map-matching integrity was in Quddus's thesis [46] and in publications published during its preparation. The author proposed a method that uses uncertainty in position, distance error and heading error to produce heuristic integrity indicator based on fuzzy inference system. He validated his integrity monitoring with three map-matching methods and reports 91% overall correct detection rate, missed detection rate lower than 8.5% and false alarm rate lower than 10.1%.

Velaga [57] developed integrity method based on the work by Quddus. The author uses two fuzzy inference systems: one is used when RAIM (receiver autonomous integrity monitoring) labels positioning fix reliable and other when not. Reported missed detection and false alarm rates are both below 1%. This yields overall correct detection rate higher than 98%.

Li et al. [36] (co-authored by Quddus) proposed a tightly-coupled integrity monitoring for map-matching. The authors review drawbacks of classical RAIM when used in the context of map-matching and then propose an integrated navigation system with adapted RAIM functionality that is better suited for this task. The navigation system integrates information from GNSS, gyroscope and digital elevation model. The integration is achieved using extended Kalman filter. Uncertainty in measurements and related filter residuals are then used in the adapted RAIM to compute the horizontal protection level (HPL). The authors tested their solution in urban and suburban field trials. They report both false and missed detection rates less than 0.1% with 99.83% overall correct detection rate. These are excellent results, however, a specialized navigation system with a gyroscope and GNSS receiver that is able to report raw pseudorange measurements is required.

Jabbour et al. [28] proposed a multiple hypothesis technique based map-matching method and a simple integrity method with it. The authors use two integrity indicators: (1) the number of effective hypotheses, N_{eff}, and (2) the "normalized innovation squared", NIS. The N_{eff} is the number of hypotheses with a high likelihood of being correct, and the NIS is similar to Mahalanobis distance between the trajectory sample and its matching in position and heading. The authors conducted a field trial with a navigation system based on GPS, odometer and fiber-optic gyroscope. Overall correct detection rate over 88.8% was reported. An adaptation of this integrity method was later used by Bonnifait et al. [9].

Toledo-Moreo et al. [56] proposed lane-level map-matching method with integrity monitoring. This solution uses an integrated navigation system based on GNSS, odometry and gyroscope. The lane-level matching is enabled via novel road-network model named *eMap* (enhanced map). The eMap models lane shapes as a clothoids (also known as Euler Spirals). Since clothoids are used in highway engineering authors argue that similar mathematical structures can be expected to emerge in sampled trajectories. The map-matching algorithm is based on a particle filter. Two integrity indicators are used: (1) lane occupancy probability (μLO) and (2) lane position protection level (LPPL). The μLO is a sum of normalized particle weights that occupy the matched lane and LPPL is positioning protection level analogous to horizontal protection level (HPL) used in the classical RAIM. The authors tested their method in three short field trials. Reported overall correct detection rate ranged from 98.7% to 99.2%. The missed detection rate was in the range from 0% to 1.2%. The false alarm rate ranged from 0.7% to 1.7%.

Integrity monitoring in map-matching is critical for many practical applications, but there are not many authors who treat this subject. The most promising results are currently by Li et al. [36] and by Toledo-Moreo et al. [56]. Both methods require navigation system that integrates information from the satellite navigation system, odometry and gyroscope.

D. Performance Evaluation

There is no consensus on how map-matching performance should be measured. In one of the early papers Bernstein and Kornhauser concluded *"it is not immediately clear what measures of performance are most appropriate or*

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

what scenarios should be evaluated" and *"in an abstract sense it is clear that we would like the algorithm to perform perfectly when the errors go to zero, but it is not entirely clear what that means in practice"* [4]. The issue authors pointed out has been preventing researches to compare their methods until today. Kubička et al. [33] published a dataset of 110 map-matched trajectories in an effort to resolve this problem. It is available online under open access [34]. It can be used for experimentation, validation and comparison as well as for machine learning and inference. The authors used freely available trip records to create a rich set of scenarios on which the community can study map-matching.

The authors often test their methods with private or proprietary data. There are exceptions, however: Newson and Krumm [42] published their testing data and described used validation methodology in detail. Their method is discussed in Section VII-C.

VII. The Methods

Many map-matching methods were proposed over the last two decades. We discuss a selection of recent innovative methods with an occasional look back to influential ideas that motivated the latest development. See [48] for a survey of the older methods.

Early methods are known as *point-to-point*, *point-to-curve*, or *curve-to-curve*. This naming convention was introduced by Bernstein and Kornhauser in "An Introduction to Map Matching for Personal Navigation Assistants" (1996) [4]. The point-to-point matching matches each sample s_i to the nearest node in V. The point-to-curve matches s_i to the nearest point on the nearest road in E. The curve-to-curve method matches the trajectory s to the most similar route r in the road-network. An example of map-matching with these methods is in Figure 5. It is based on the map and the trajectory from Figure 3. In Figure 5(a) the point-to-curve method matches trajectory samples (squares) to their respective closest points on the road-network (circles). In Figure 5(b) the curve-to-curve method matches the geometric shape of a candidate path (blue curve) to the geometric shape of the trajectory (red curve). Historically, the point-to-curve methods lead the development of online map-matching while curve-to-curve methods lead the research on offline methods.

White et al. [66] experimented with four basic map-matching methods. First is classical point-to-curve, it ignores past samples and makes no effort to make the resulting route contiguous. The second method is a modified version of the first one where vehicle heading is taken into consideration. The third method is based on the second but additionally considers road-network topology. The fourth method is the curve-to-curve matching. The results are summarized in Table 2. It was tested on four routes; we report worst-to-best performance range for each. The percentage values represent the correctly matched samples with respect to the number of samples in the trajectory. The authors found that the enhanced point-to-curve methods perform better than curve-to-curve matching. They attributed it to high sensitivity to outliers of the latter.

A. Geometric

Given some trajectory s the geometric methods search for the most resembling route in the map using some shape similarity metric δ. Consider a polygonal curve $r = (\mathrm{loc}(v_1), \mathrm{loc}(v_2), \ldots)$ that represents the geometric shape of the path $(v_1, v_2, \ldots)$ in G and consider $\mathcal{R}$ a set with shapes of all paths in G. Then, geometric methods conduct a search for the path in $\mathcal{R}$ that maximizes its similarity δ to the trajectory s. Hence, a generic model of geometric methods reads

$$r = \underset{x \in \mathcal{R}}{\operatorname{argmax}}\, \delta(s, x) \tag{2}$$

A typical application is in offline settings. This makes geometric methods suitable for mapping and tracking. Online map-matching is possible but computationally demanding.

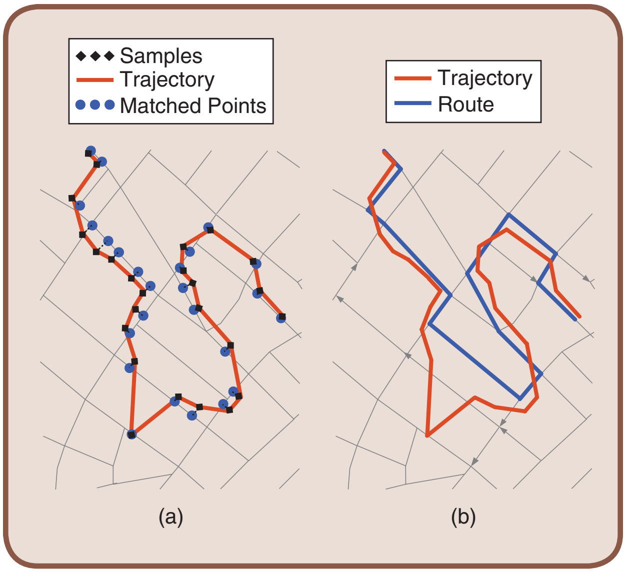

FIG 5 Basic map-matching methods. (a) point-to-curve, (b) curve-to-curve.

Table 2. Base Line Performance.

Method	Matching Accuracy[1]
point-to-curve	53 – 67%
point-to-curve, considers heading	66 – 85%
point-to-curve, enforces route contiguity	66 – 85%
curve-to-curve	61 – 72%

[1]based on White et al. [66].

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

Sparsely sampled trajectories are challenging for these methods since they are based on a hypothesis that the trajectory is geometrically similar to the correct route. The trajectory contains less detail when sampled sparsely. Nevertheless, there are successful applications of geometric methods on sparse trajectories. As dicussed below, this is achieved by combining purely geometrical approach with heuristics that help to resolve ambiguities.

Two similarity metrics received attention in the literature: the *Hausdorff distance* and the *Fréchet distance*. The *one-sided Hausdorff distance* from curve A to curve B is defined as

$$\delta'_H(A,B) = \max_{a \in A} \min_{b \in B} d(a,b) \tag{3}$$

where $d(a,b)$ is the distance between a and b. The metric is either the Euclidean distance or the great-circle distance. The Euclidean distance is suitable on short ranges. The great-circle distance is used when the curvature of

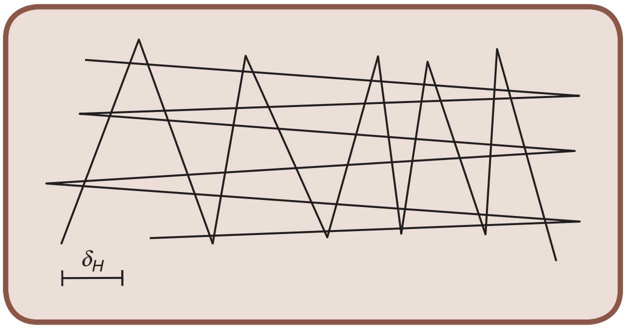

FIG 6 Two curves and their Hausdorff distance. The δ_H indicates the Hausdorff distance in scale (adaptation of example given by Alt and Godau in [2]).

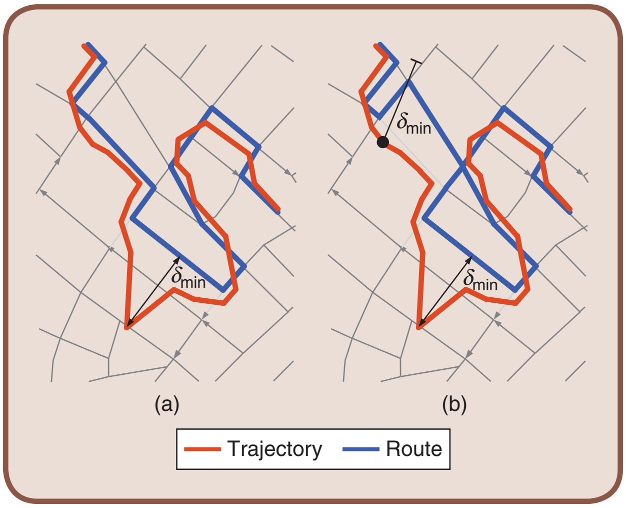

FIG 7 An example of sensitivity to outliers with Fréchet distance based geometric map-matching: both routes have the same Fréchet distance δ_{min} to the trajectory. (a) correct route, (b) incorrect route.

Earth becomes a concern. The Haversine or Vincenty's [58] formulæcan be used.

The Hausdorff distance δ_H is defined as the maximum of the two one-sided Hausdorff distances:

$$\delta_H = \max(\delta'_H(A,B), \delta'_H(B,A)) \tag{4}$$

Its computation on polygonal lines is straightforward from the definition in $O(nm)$ time, with n being the number of samples in the trajectory and m the number of nodes in the route. The asymptotic run time $O((n+m)\log(n+m))$ can be achieved if Voronoi diagrams are used to answer the nearest neighbor queries faster [3].

The Hausdorff distance ignores the course of the two curves. Consider two mutually overlapping curves in reversed order (i.e. for curve $A = (a_1, a_2, \ldots a_n)$ the $B = (a_n, a_{n-1}, \ldots, a_1)$. The Hausdorff distance between them is zero by definition. Consequently, a map-matching method that uses the Hausdorff distance cannot distinguish between two routes that go on the same roads in mutually opposite directions. In general, any two curves that occupy the same area can have small Hausdorff distance even if the two shapes are wildly different. See Figure 6 for one such example. This is why the Hausdorff distance is seldom used in geometric map-matching.

The Fréchet distance is more common. It was introduced in Maurice Fréchet's thesis "Sur quelques points du calcul fonctionnel" in 1906 [20]. The definition reads

$$\delta_F(f,g) = \inf_{\alpha,\beta} \max_{t \in [0,1]} d(f(\alpha(t)), g(\beta(t))) \tag{5}$$

where f, g are parametrizations of the two curves $(f,g:[0,1] \to \mathbb{R}^2)$ and α, β are continuous, monotone, increasing reparametrizations $(\alpha,\beta:[0,1] \to [0,1])$. The reparametrization functions are introduced to enforce continuous and monotonically increasing parameters for f and g. Famous illustration goes as follows: suppose a man is walking his dog, both the man and the dog walk along their own trajectory. The maximum length of the leash is then equal to the Fréchet distance between trajectories of the man and his dog.

Map-matching methods based on both Hausdorff and Fréchet distances are sensitive to outliers. This is due to the maximization operator in definitions (4), (5). See Figure 7 for an example. The Figure 7(a) shows the trajectory and correctly matched route. Figure 7(b) shows incorrectly matched route. Both routes have the same Fréchet distance δ_{min} to the trajectory due to an outlier in another part of the trajectory (the sample s_{11}; see Figure 3(c)). Since both these routes have minimal Fréchet distance to the trajectory the map-matching algorithm is by definition (2) indifferent between them. As can be seen in Figure 7(b) this can result in obviously wrong matching.

In order to compute the Fréchet distance one has to identify the reparametrization functions α, β such that the

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

maximum distance between the two curves f, g is minimized. This has proven difficult, no algorithm for continuous curves has been found [3]. Alt and Godau [2] discovered an algorithm to compute the Fréchet distance between two polygonal curves. It runs in $O(nm \log^2(nm))$, where n and m are cardinalities of the two curves.

Cao and Wolfson [11] used the Hausdorff distance for "nonmaterialized trajectory representation". The authors formulate map-matching as a spatial mismatch correction problem and assume that the vehicle is always on the road to solve it. Their algorithm adjusts the trajectory to the road-network such that the Hausdorff distance between the route and the trajectory is minimized. The strength of their contribution is that their solution is exact with respect to their problem formulation. The practical application is likely limited due to properties of the Hausdorff distance discussed above. The asymptotic complexity of the algorithm is $O(n|E|^2)$, where E is the set of arcs in the road-network graph G as introduced in Section IV.

Alt et al. published first map-matching method based on the Fréchet distance [1]. The authors first solve decision problem whether $\delta_F(f, g) \leqslant \varepsilon$, for some $\varepsilon > 0$ and then use the solution to find the minimal ε. The method is optimal in the sense that it finds a route whose Fréchet distance to the trajectory is minimal. Nevertheless, if there are outliers in the trajectory, then there can be multiple routes with minimal Fréchet distance as discussed above. The computational demand of the algorithm is asymptoticaly bounded by $O(n|E|\log(n|E|)\log(|E|))$.

Brakatsoulas et al. [10] followed on the work of Alt et al. and introduced a *weak Fréchet distance* in an attempt to deal with the high computational demand of the original. The weak Fréchet distance[3] is a relaxed version of the Fréchet distance that does not impose the non-decreasing property on reparametrizations $\alpha(\cdot)$ and $\beta(\cdot)$. It is asymptotically faster (runs in $O(n|E|\log(n|E|))$ time) and, according to authors, produces results similar to (5), most of the time. This, however, is not guaranteed. It is possible to find two curves whose weak Fréchet distance is different from the Fréchet distance.

Another proposition to solve the map-matching problem using the weak Fréchet distance is by Wenk et al. [65]. Their "adaptive clipping" algorithm speeds up the weak Fréchet distance computation. The asymptotic run time is slightly worse than the run time of the method proposed by Brakatsoulas et al. The authors, however, report that average run time is reduced significantly.

Further speedup of the Fréchet distance computation was reported by Chen et al. [12] (co-authored by Driemel and Wenk). The authors extend the results of Driemel et al. [18] for Fréchet distance approximation in near linear time. The authors report speedup with respect to Alt et al. [1] on the order of 10^3, in some cases.

[3]In some sources referred to as nonmonotonic Fréchet distance.

Wei et al. [62] revisited the problem with the sensitivity to outliers. The authors first find all routes with minimal Fréchet distance to the trajectory. If there is a unique solution then it is returned as a correct match. A fitness function is used to find the most likely match if there are multiple solutions. Its performance depends on the choice of the fitness function. It is often based on empirical formulas that can be biased to work in the environment they were identified in. Still, this is a sophisticated approach with outlook to perform well in terms of accuracy and speed if used in conjunction with near linear time approximation of the Fréchet distance (proposed by Driemel et al. [18] and adapted for map-matching by Chen et al. [12], see above).

In summary, Alt et al. [1] pioneered the geometric map-matching with their method to compute the Fréchet distance. The authors Brakatsoulas et al. [10], Wenk et al. [65] and Chen et al. [12] contributed to significant advances in lowering the computational requirements of the Fréchet distance based map-matching. However, sensitivity to outliers, which is the main drawback of this technique, remained unresolved. Wei et al. [62] introduced a combined approach that uses a heuristic fitness function to solve this issue.

The performance of geometric map-matching was studied by Wei et al. [63]. The authors compare their method to the state-of-the-art and show that their approach outperforms other methods on low sampling rates and has consistently lower mismatching rate. In particular, they have shown superior performance with respect to popular methods by Newson et al. [42] and Lou et al. [38]. Nevertheless, recent advances contributed by Hunter et al. [27] are likely to be superior. Hunter et al. proposed a framework that considers additional information such as the agreement between traveled distance according to the matched route and according to the sampled trajectory.

B. Multiple Hypothesis Technique

The multiple hypothesis technique (MHT) was originally developed for object tracking. A number of its variations were used in the context of map-matching as well. Multiple hypothesis methods make use of map topology to infer where the vehicle might have gone from its prior positions. This has a number of advantages: it enforces route contiguity and enables a recursive solution of the map-matching problem. However, the set of hypotheses can grow exponentially in the worst case.

As the name suggests, these methods maintain a set of *hypotheses*. The term hypothesis is used as a synonym to any candidate route under consideration. A set of seed hypotheses is generated from the first sample, usually with the point-to-curve method. Then as further samples arrive the method updates its set of hypotheses and estimates the likelihood of each being the correct route. The updating process consists of *hypothesis branching* and *hypothesis pruning*. A hypothesis is branched when the vehicle arrives

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

at a crossroad: the original (parent) hypothesis is replaced with new (child) hypotheses. Each child is a clone of the parent extended into one of the directions the vehicle can take on the crossroad. This guarantees that there will always be a hypothesis spanning the whole trajectory. Hypothesis pruning is used to remove outdated hypotheses. Three pruning criteria are currently used by authors: (1) limit the maximum number of hypotheses; (2) threshold likelihood scores; (3) keep only those hypotheses that are close to the latest position.

The multiple hypothesis paradigm can be formulated recursively: partial results can be obtained before the trajectory is fully processed. Each sample can be processed in time that scales linearly with the number of hypotheses, while the number of hypotheses is usually kept low due to hypothesis pruning. These properties make it particularly suitable for online map-matching and navigational applications. Another advantage is that basic failure detection is implicit: if there are no hypotheses, then some failure must have occurred.

Pyo et al. [45] designed a method for robust online map-matching. The authors derive the probability of each hypothesis using the Bayes rule, asserting all hypotheses to be true and then computing the probability of the assertion being correct for each. The pruning is done by thresholding the probabilities: the hypothesis gets pruned if it gets below some predefined threshold. The authors also threshold the probabilities to decide whether the hypothesis is tentative or confirmed. This serves as simple integrity monitoring. Finally, authors always add an "off road" hypothesis between the candidates to account for the case when the road is missing in the map. The method was validated in a field test. The authors used GPS augmented with deduced reckoning based on a gyroscope and odometer. The method has failed to decide on correct matching in 4–12% cases depending on the context, but no mismatchings were observed. It also failed to identify off road condition 17% of the time, but it never matched the trajectory to an incorrect road.

Marchal et al. [39] developed a simple and fast multiple hypothesis technique based method for offline map-matching. The authors use simple scoring based on the distance between individual samples and their matched counterparts on the hypothesized routes. The hypotheses are branched when the vehicle gets to the second half of current road. The list of candidate hypotheses is limited to N most likely hypotheses, where the N is chosen beforehand. The hypothesis with the highest score is returned as the matched route. The method was validated in a field test. The authors report that 3.3% of the samples were matched to roads more distant than the accuracy of the map (10 meters per point) while the computation time was 10^3 times faster than the sampling period of the GPS receivers. This method is simple and the reported accuracy can be considered sufficient for some applications. The authors Schuessler and Axhausen [52] followed on work of Marchal et al. They observed that the original does not perform well when there are two parallel roads close to each other. They adapted the scoring function such that differences between the vehicle speed and the speed limit are punished with lower scores in response to that. They report that this helps to discern the position of the vehicle if the speed limits are different on two parallel roads.

Kubička et al. [35] developed a method for online map-matching using multiple hypothesis technique. The authors use *gating technique* for branching and pruning. The gate is defined as a spherical area around the latest reported position with a radius larger than maximum positioning error. Hypotheses are branched whenever a crossroad is found within the gate and pruned whenever hypothesis drops out from the gate. This guarantees that correct hypothesis will not get pruned. The authors compare the travel distance reported by the positioning system with the distance traveled along the hypothesized routes. The difference is considered in the scoring function together with average positioning error between the trajectory and the route. The method was validated in two short campaigns in rural and urban environments. The authors did not observe any error on rural roads. In urban areas they observed 0.5% samples temporarily mismatched on crossroads.

The interest in multiple hypothesis technique based methods spans from their simplicity. According to reported results, the recent methods might be able to offer a good trade-off between matching accuracy and computational demand. For example, Marchal et al. [39] contributed a method that is too simplistic to compete with the state of the art, but the reported accuracy might be sufficient for some applications. Kubička et al. [35] proposed an online map-matching method with low computational demand and with a guarantee that correct hypothesis is never pruned. This method is suitable for map-matching with purely satellite-based positioning. Pyo et al. [45] contributed with a method suitable for online map-matching in built-in vehicle navigation assistants that have access to vehicle odometry and gyroscope data.

The multiple hypothesis technique based methods compete with other online methods, most prominently with fuzzy logic based methods [21], [29], [31], [47], [54]. Performance comparison between them is difficult as different authors use different methodologies to evaluate it. It is nevertheless reasonable to expect that fuzzy logic based method tuned specifically to the navigation system in use will show superior performance over multiple hypothesis technique based methods. The multiple hypothesis technique based methods, on the other hand, are simple, require little tuning and offer accuracy sufficient for some applications.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

C. Hidden Markov Models and Conditional Random Fields

The map-matching based on hidden Markov models received considerable attention in connection with tracking applications. The first method that uses it was published in 2006 by Hummel [26]. Over ten other methods were proposed since, many are well cited. The majority of them is designed for offline map-matching. Online map-matching is possible. However, the matching is usually limited to last few samples to reduce the computational effort. This is known as the *sliding window technique* [22].

To shortly review the classic theory, a Markov chain is a stochastic model of a system that can randomly change state such that probability of the next state depends only on the current state. The set of states is finite. It is often visualized with a graph where nodes represent states and arcs represent transitions between them. Each state defines *transition probability* distribution over the set of states. It describes the probability of transition from the current state to any other state. hidden Markov model is a generalized Markov chain where the current state is not directly observable. The relation between observations and internal states is described using *emission probability* distribution $p(l|x)$. It is a conditional distribution that the system is in the state l if x was observed. In the context of map-matching, the x is the observed position while the l is the road on which the vehicle might have been when the measurement took place. See [49] for more information on hidden Markov models.

Most methods model emission probabilities with a model that considers only the distance between the trajectory and the road. Positioning errors are assumed to follow isoradial Gaussian distribution. Probability $p(x|l)$ (not to be confused with $p(l|x)$ of observing some position x given that the vehicle is on road l is

$$p(x|l) \approx \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-d^2}{2\sigma^2}} \tag{6}$$

where d is minimum distance between observation x and any position on the road l. The emission probability $p(l|x)$ can be obtained with the help of the Bayes rule. It simplifies to

$$p(l|x) = \frac{p(x|l)p(l)}{\sum_{k=1}^{n_l} p(x|k)p(k)} \tag{7}$$

where n_l is the number of roads in the road-network and $p(l)$ is the prior probability of being on the road l (prior distribution). Most methods don't have any upfront preference about the road on which the vehicle might be. If that is the case, then the prior distribution is uniform over the set of candidate roads.

With the hidden Markov model set up, we can perform inference on it. We are interested in the most likely sequence of states (that is, the route r), given a sequence of observations (that is, the trajectory s). The standard method to compute it is with the *Viterbi algorithm* [60]. Its asymptotic run time is $O(nm^2)$, where n is the number of observations (trajectory samples) and m is the number of states (roads). All map-matching methods discussed below have their run time asymptotically dominated by the run time of the Viterbi algorithm.

Hummel [26] developed a Bayesian classifier to match a single trajectory sample and used the hidden Markov model to identify the route with it. The method was designed for online applications. Author identifies most likely road on which the vehicle travels using a search for road that minimizes Mahalanobis distance

$$\delta_M = \left(\frac{d}{\sigma_d}\right)^2 + \left(\frac{\delta\phi}{\sigma_\phi}\right)^2 \tag{8}$$

where d and $\delta\phi$ are positioning and heading errors between the sample and the road. Parameters σ_d and σ_ϕ are standard deviations in position and heading respectively. The hidden Markov model structure follows the structure of the road-network: Markov states represent roads and transitions represent the turns the vehicle can take on an intersection. Transition probabilities are distributed uniformly between the turns the vehicle can make on crossroads. See Figure 8 for comparison between the road network graph and corresponding hidden Markov model. Note the self-transitions on each state: they allow the

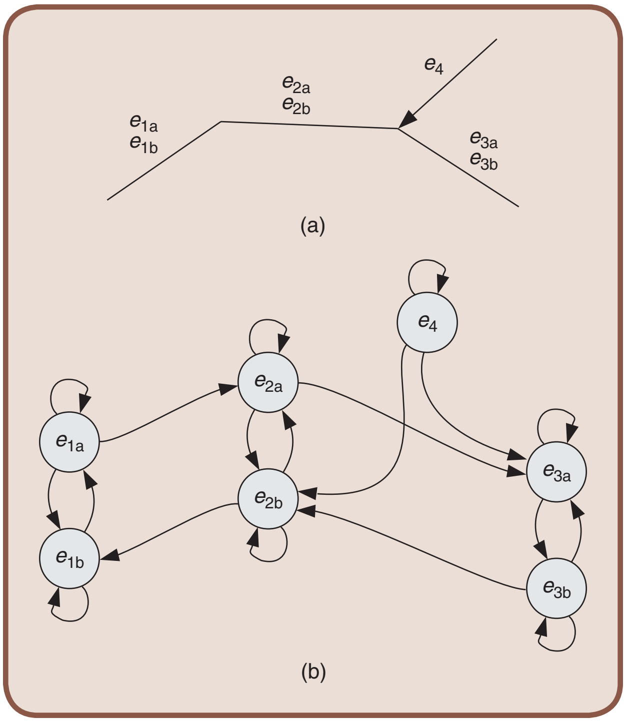

FIG 8 Markov model structure according to Hummel [26] and Pink and Hummel [44]. (a) road network (arcs e_1, e_2, e_3 are bidirectional), (b) corresponding Markov model structure.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

vehicle to stay on the same road for a time. The dependence on vehicle heading makes this method robust, but it can have the opposite effect in some situations:

- Uncertainty in heading can be significant when moving slowly (see Figure 4(a)).
- The heading errors can be artificially high (up to 45 degrees) when performing a sharp turn on an intersection

The method was validated with a field test in an urban area. Duration of the test was roughly four hours. The trajectories were collected using a low-cost GPS receiver at 1 Hz sampling rate in Karlsruhe, Germany. All trajectories were matched correctly, although authors report that 0.4% of samples were temporarily mismatched. These errors occurred in short duration until the belief in the correct route built up sufficiently. They can be considered as mismatches in online map-matching, but they have no effect on offline map-matching performance.

Pink and Hummel [44] extended results by Hummel [26]. The authors introduced a number of innovations with respect to the original method. They preprocessed the trajectory with a Kalman filter in order to suppress outliers (the filter is discussed in section VI-B). Another innovation was in road network modeling: authors used a cubic spline interpolation to generate continuous curved paths. It models the trajectory shapes conventional vehicles normally follow. This was motivated by the problems with the sensitivity of (8) to heading errors. This method is designed to be robust against positioning errors, however, authors do not consider trajectories with missing samples due to lacking satellite signal. This makes it suitable for mapping applications where the positioning system is able to provide the position fixes without interruption.

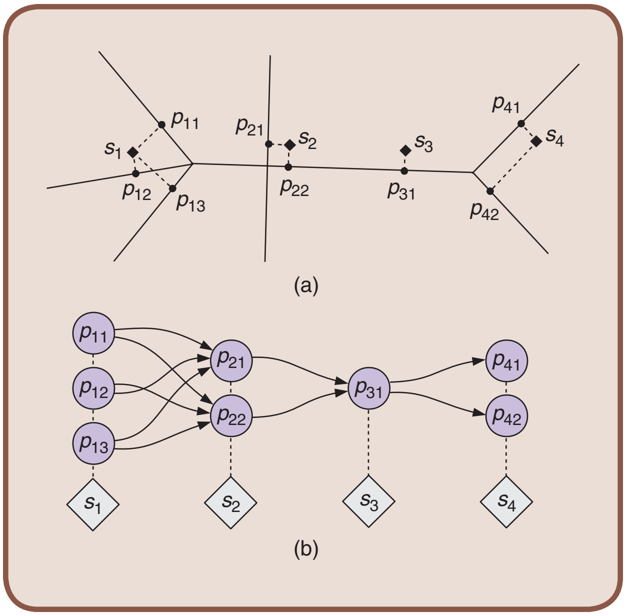

FIG 9 hidden Markov model according to Krumm et al. [32] and Newson and Krumm [42]. An example. (a) situation, (b) corresponding hidden Markov model.

Krumm et al. [32] developed an offline map-matching method that constrains the solution to those routes whose expected travel time is congruent with observed travel time. The authors used a novel structure of the hidden Markov model to achieve this. Instead of recycling road-network topology as Hummel did, Krumm et al. build their Markov model from trajectory data. See Figure 9 for an example. A set of states that represent possible positions of the vehicle on the map is created for each sample. Let us denote this set P_i for sample s_i and let us denote j-th matching in P_i as p_{ij}. The authors allow only transitions from positions in the set P_i to the set P_{i+1}. The transition probability from some $p_{ij} \in P_i$ to $p_{(i+1)k} \in P_{i+1}$ is estimated according to the agreement between observed and expected travel times. The observed travel time is the time difference between the two samples. The expected travel time is estimated using the average speed along the roads on the shortest path between the two candidate positions p_{ij} and $p_{(i+1)k}$. Emission distributions are based on the model (7). The authors have shown on numerous examples that their method is able to deal with difficult matching scenarios but did not quantify its performance.

Newson and Krumm [42] argued that estimated travel time might be affected by immediate traffic on the road. They addressed this issue and extended the method of Krumm et al. [32]. They changed the way transition probabilities are modeled: instead of working with travel times authors make use of distances. They compare the great-circle distance between two consecutive samples with the distance along the road-network. The routes with minimial difference between the two are preferred. Another novelty introduced by the authors is preprocessing. They downsample the trajectory such that samples within two standard deviations from the last accepted sample are ignored. Median absolute deviation [23] is used to estimate the standard deviation. It is a robust measure of variability in univariate data set defined as

$$\mathrm{MAD} = \mathrm{med}_i \left| X_i - \mathrm{med}_j(X_j) \right| \tag{9}$$

where X is the data set whose variability we estimate. It has to be scaled in order to obtain a consistent estimator of standard deviation. This depends on distribution type. The authors assumed the positioning error follows the Gaussian distribution for which $\sigma \approx 1.4826\mathrm{MAD}$. They tested the method on 80-kilometer trajectory collected in Seattle, US. The performance was evaluated in a novel way by comparing total length of the roads that were matched incorrectly to the length of those matched correctly. The error is computed as

$$e = \frac{d_+ + d_-}{d_0} \tag{10}$$

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

where d_0 is the length of correctly matched roads, d_+ the length of mismatched roads that were added to the route and the d_- length of mismatched roads that were missing in the route. The authors report no errors on sampling periods shorter than 30 seconds. Wei et al. [63] tested this method independently and observed 98.5% matching accuracy on one-second sampling period and 95% matching accuracy on 60 second period.

Hunter et al. [27] developed a "path inference filter". The authors show that map-matching based on hidden Markov models is subject to selection bias and propose the path inference filter as the solution. It is based on conditional random fields, a generalization of hidden Markov models. The authors build directed graph with similar structure as the model used by Krumm et al. (Figure 9(b)). A set P_i of candidate positions on the road-network is created for each sample s_i in the trajectory. Each candidate position $p_{ij} \in P_i$ has assigned a score that represents the likelihood that vehicle was in position p_{ij} when s_i was observed. The probabilistic model (7) is used for this. The authors create arc for each path the vehicle can take between each pair p_{ij} and $p_{(i+1)k}$ in $P_i \times P_{i+1}$. Each such path has assigned a score according to a *driver model* that represents the likelihood that vehicle took this path. The number of paths for each pair in $P_i \times P_{i+1}$ is upper bounded by speed limits on the roads in question. The authors list all complete paths in this graph (candidate routes from origin to destination) and define *potential* of each as a product of the scores of the candidate positions and of the candidate routes between them. The potential function becomes probability mass function when normalized to one. The authors use the Viterbi algorithm to identify the most probable route using the normalized potentials. The path inference filter was primarily developed for tracking applications. The authors report that it stays competitive over the full range of sampling rates. Testing has shown ability to match all routes correctly on trajectories with high sampling rates and 75% of the trajectories with two-minute sampling period.

The hidden Markov model based map-matching was successfully applied on both online and offline map-matching. The online method by Pink and Hummel features exceptional robustness against positioning errors, but it is not robust against missing trajectory samples. The offline method by Newson et al. shows high matching accuracy over a wide range of sampling rates. Its accuracy is comparable to state-of-the-art geometric method [62] with simpler apparatus and with lesser computational demand. However, recently published path inference filter by Hunter et al. [27] is likely to outperform it. Although computationally demanding, this is a sophisticated method that integrates spatial and temporal relationships as well as driver behavior elegantly in a conditional random field-based model.

The methods discussed in this section also introduced a number of techniques that were reused by other authors. The modeling approach introduced by Krumm et al. [32] was adapted and extended later by Lou et al. [38], Newson et al. [42] and Hunter et al. [27]. Also, the technique of Newson et al. who compare expected and observed travel distance was later reused by Kubička et al. [35] and by Hunter et al.

VIII. Method Selection

Different map-matching methods are suitable for different map-matching applications. There is no universal method that would suit the needs of all. In this section, we consider the trade-offs that must be made when selecting a map-matching method. We consider map-matching performance, required computational effort, sensitivity to tuning, whether integrity monitoring is needed and whether preprocessing is needed.

The navigational applications require online, high sampling rate map-matching methods. The computational effort must be kept low as the system is required to respond in real time. When integrity monitoring is needed, then the method by Toledo-Moreo et al. [56] should be considered. This method has shown high matching accuracy on lane-level while providing continuous integrity monitoring of the matching output. Methods based on fuzzy logic, such as [47], have reportedly excellent matching accuracy but require expert knowledge for their tuning. Similarly, methods based on belief theory [19], [41] are difficult to tune, but can reportedly match accurately. The methods based on multiple hypothesis technique [35], [45] might be able to offer an interesting trade-off between computational demand and matching accuracy. The Hidden Markov model based methods and geometric methods are not well suited as they require considerable computational resources. The sliding window technique can be used to remedy this issue: only a last few samples are used to map-match the current matching point. When the demand on computational effort is not stringent, then the method by Hummel [26] should also be considered as it has outlook to be robust against positioning errors.

The tracking applications require offline, low sampling rate map-matching methods. Higher computational effort can be tolerated as the trajectories are postprocessed after they were collected. The most advanced method in terms of matching accuracy is reportedly the path inference filter [27]. However, its computational demand might be too prohibitive. The method by Newson et al. [42] offers good matching accuracy while its computational demand is comparatively low. Another option is the geometric method by Wei et al. [62], especially when used in conjunction with fast Fréchet distance approximation method developed by Driemel et al. [18]. If a massive set of trajectories must be processed and matching accuracy is not critical,

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

then the method by Marchal et al. [39] can be considered. If the application makes use of sparsely sampled trajectories, then low-sampling rate methods [13], [38], [50], [71] can be of interest. These methods are likely to be outperformed by the path inference filter, but they are often easier to implement.

The mapping applications are similar to tracking applications in the sense that they require offline methods. However, the trajectories are sampled densely. The positioning system needs to be accurate as sampled trajectories are used to measure road shape. It is used when introducing new roads to the map. The matching accuracy is critical, while the computational effort is not. The geometric methods are well suited for this [1], [10]. While they are sensitive to outliers, this is not critical when accurate positioning system is used. The method by Pink and Hummel [44] also has interesting properties with respect to mapping applications: it uses Kalman filter based preprocessing and a hidden Markov model that has shown high matching accuracy while having lower computational demand. The Kalman filter can be used separately in connection with some other map-matching method.

IX. Summary & Conclusions

This work reviews map-matching methods according to their intended application. This is motivated by the fact that the map-matching problem is not unique. Nature of each specific problem implies a different approach. The separation according to the intended application conveniently resolves the ambiguity in the definition of the problem. To the author's best knowledge there are two precedent reviews on map-matching. There has been significant development on the subject since the first review (published in 2007). The second review is more recent, from 2014, but is limited to a subset of map-matching methods. Both reviews did not consider the categorization according to the intended application. We hope that our contribution will help the community to understand the map-matching problem better and to navigate in the plethora of methods available today.

Three broad application areas were identified: *navigation*, *tracking* and *mapping*. Each of them has its own specificities. Most prominently, map-matching for navigation matches individual samples to the road-network while tracking and mapping methods match complete trajectories to complete routes. Most published methods describe themselves as either *online or offline* and with *low or high sampling rate*. Navigational methods are typically online, high sampling rate methods while tracking and mapping methods are typically offline.

Four research challenges were discussed. Dealing with errors in positioning data and in maps is the original challenge in map-matching. Pure satellite-based positioning can suffer a temporary lack of data due to limited satellite visibility. Maps can feature a mixture of random, systematic and modeling errors that are difficult to account for. The second research challenge is integrity monitoring. Errors in both positioning data and maps are typically unbounded. This implies that there can never be any guarantee of correct matching. It also implies that undetectable missed integrity alarms are possible. The third discussed research challenge is trajectory preprocessing. These techniques are reportedly often used in mapping applications and sometimes in tracking, yet literature on it is scarce. The published techniques are reviewed, and their properties are discussed. Finally, map-matching performance evaluation is listed as the last research challenge. It addresses the lack of consensus on how map-matching performance should be evaluated.

A selection of map-matching methods is reviewed. The methods are categorized by their scientific basis and presented in such an order that contribution and impact of each is recognized and placed in context. Each method is also classified and linked to its applications. Geometric, multiple hypothesis technique, hidden Markov model and conditional random field based methods are discussed. The geometric methods are suitable for mapping where high sampling rate from uninterruptible positioning system is used. Their weakness is their sensitivity to outliers. They can be used for tracking with trajectories sampled at lower sampling rates. However, their performance is likely inferior with respect to other recent methods. The multiple hypothesis technique based methods are suitable for online map-matching, hence for navigational applications. According to reported results, the recent methods might be able offer a good trade-off between matching accuracy and computational demand. Their other advantage is that they require relatively little tuning when compared to methods based on, for example, fuzzy logic or belief theory. The hidden Markov models were successfully applied on both online and offline map-matching. They are often used in tracking applications. They feature high accuracy over a wide range of sampling rates. While the accuracy is reportedly comparable to some geometric methods, these methods use simpler models and do not rely on heuristics. Finally, the path inference filter seems to be a promising step beyond the limitations of the hidden Markov model based methods. Its authors formulate an argument that hidden Markov model has selection bias problem when used in map-matching and propose the path inference filter that avoids it.

The review is limited to outdoor vehicular map-matching. Indoor map-matching as well as SLAM (simultaneous localization and mapping) is not covered in this review. A number of methods which fit in this scope was not covered as well. Instance of these are fuzzy logic based methods and belief theory based methods. They are covered in previous reviews.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

Acknowledgments

The authors would like to thank the associate editor and anonymous reviewers for their comments that helped us to improve the overall quality of this paper.

About the Authors

Matěj Kubička obtained his engineering degrees in 2014 from West Bohemia University in Pilsen, Czech Republic and from ESIEE Paris in France. He is currently a Ph.D. student at the Laboratory of Signals and Systems (a joint research unit of CNRS, CentraleSupelec and Université Paris-Sud) and at IFP Energies Nouvelles, France. His research activities focus on intelligent transportation systems, especially on green transportation, eco-routing and map-matching.

Arben Cela received the B.S. and Ing. degrees from the Polytechnic University of Tirana, Albania in 1984. From 1984 to 1986 he was with the Department of Power Generation of Faculty of Engineering of Polytechnic University of Tirana, Albania. He received the Ph.D. degree and French Habilitation (HDR) from Université Paris-Sud respectively in January 1993 and in June 2011. He is currently with the Laboratory of Signals and Systems (L2S, UMR CNRS 8506), CNRS-CentraleSupélec, Gif-Sur-Yvette 91192, and with Computer Science and Telecommunication Department of ESIEE Paris, UPE, Noisy-Le-Grand 93162, France. He is author/coauthor of one book and of more than 85 conference and journal papers in the area of optimization, control of distributed real time systems and delayed systems.

Hugues Mounier obtained his Ph.D. in automatic control in 1995 from the Laboratoire des Signaux et Systèmes of the Université Paris-Sud, Orsay, France. From 1998 to 2010 he has been with the Institut d'Electronique Fondamentale of the same University. He is currently Professor at the Laboratoire des Signaux et Systèmes. His research interests include flatness based infinite dimensional systems with applications to neuroscience, biophysical and automotive systems.

Silviu-Iulian Niculescu received the B.S. degree from the Polytechnical Institute of Bucharest, Romania, the M. Sc., and Ph.D. degrees from the Institut National Polytechnique de Grenoble, France, and the French Habilitation (HDR) from Université de Technologie de Compiègne, all in Automatic Control, in 1992, 1993, 1996, and 2003, respectively. He is currently Research Director at the CNRS (French National Center for Scientific Research) and the head of L2S (Laboratory of Signals and Systems), a joint research unit of CNRS with CentraleSupélec and Université Paris-Sud located at Gif-sur-Yvette. He is author/coauthor of 5 books and of more than 400 scientific papers. His research interests include delay systems, robust control, operator theory, and numerical methods in optimization, and their applications to the design of engineering systems.

References

[1] H. Alt, A. Efrat, G. Rote, and C. Wenk, "Matching planar maps," in *Proc. ACM-SIAM 14th Annu. Symp. Discrete Algorithms*, 2003, pp. 589–598.
[2] H. Alt and M. Godau, "Computing the Fréchet distance between two polygonal curves," *Int. J. Comput. Geometry Applicat.*, vol. 5, no. 01, pp. 75–91, 1995.
[3] H. Alt and L. J. Guibas, "Discrete geometric shapes: Matching, interpolation, and approximation," *Handbook Comput. Geometry*, vol. 1, pp. 121–153, 1999.
[4] D. Bernstein and A. Kornhauser, "An introduction to map matching for personal navigation assistants," TIDE Center, NJ, Tech. Rep., 1996.
[5] M. Bierlaire, J. Chen, and J. Newman, "A probabilistic map matching method for smartphone GPS data," *Transport. Res. Part C Emerging Technol.*, vol. 26, pp. 78–98, 2013.
[6] M. Bierlaire, J. Newman, and J. Chen, "A method of probabilistic map distribution of path likelihood," in *Proc. 9th Swiss Transport Res. Conf.*, Ascona, 2009.
[7] Transportation Research Board, *NCHRP Synthesis 301: Collecting, Processing, and Integrating GPS Data into GIS*. Washington, DC: National Academy Press, 2002.
[8] P. Bonnifait, M. Jabbour, V. Cherfaoui, G. Dherbomez, and J. Laneurit, "Multi-hypothesis map-matching integrity for cooperative systems," in *Proc. European Navigation Conf.*, 2008, pp. 23–25.
[9] P. Bonnifait, J. Laneurit, C. Fouque, and G. Dherbomez, "Multi-hypothesis map-matching using particle filtering," in *Proc. 16th World Congress ITS Systems Services*, 2009, pp. 1–8.
[10] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *Proc. 31st Int. Conf. Very Large Data Bases*, 2005, pp. 853–864.
[11] H. Cao and O. Wolfson, "Nonmaterialized motion information in transport networks," in *Proc. Int. Conf. Database Theory*, 2005, pp. 173–188.
[12] D. Chen, A. Driemel, L. J. Guibas, A. Nguyen, and C. Wenk, "Approximate map matching with respect to the Fréchet distance," in *Proc. SIAM Algorithm Engineering and Experiments Conf.*, 2011, pp. 75–83.
[13] F. Chen, M. Shen, and Y. Tang, "Local path searching based map matching algorithm for floating car data," *Proc. Environ. Sci.*, vol. 10, pp. 576–582, 2011.
[14] J. Chen and M. Bierlaire, "Probabilistic multimodal map matching with rich smartphone data," *J. Intell. Transport. Syst.*, vol. 19, no. 2, pp. 134–148, 2015.
[15] W. Chen, Z. Li, M. Yu, and Y. Chen, "Effects of sensor errors on the performance of map matching," *J. Navig.*, vol. 58, no. 2, pp. 273–282, 2005.
[16] P. Davidson, J. Collin, J. Raquet, and J. Takala, "Application of particle filters for vehicle positioning using road maps," in *Proc. 23rd Int. Technical Meeting Satellite Division Institute Navigation*, Portland, OR, 2010, pp. 1653–1661.
[17] F. van Diggelen, "GNSS accuracy: Lies, damn lies and statistics," *GPS World*, vol. 18, no. 1, Jan. 2007.
[18] A. Driemel, S. Har-Peled, and C. Wenk, "Approximating the Fréchet distance for realistic curves in near linear time," *Discrete Comput. Geometry*, vol. 48, no. 1, pp. 94–127, 2012.
[19] T. Feng and H. J. P. Timmermans, "Map matching of GPS data with Bayesian belief networks," *J. Eastern Asia Soc. Transport. Stud.*, vol. 10, pp. 100–112, 2013.
[20] M. R. Fréchet, "Sur quelques points du calcul fonctionnel," *Rendiconti del Circolo Matematico di Palermo (1884–1940)*, vol. 22, no. 1, pp. 1–72, 1906.
[21] M. Fu, J. Li, and M. Wang, "A hybrid map matching algorithm based on fuzzy comprehensive judgment," in *Proc. Int. IEEE Conf. Intelligent Transportation Systems*, 2004, pp. 613–617.
[22] C. Y. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet, "Online map-matching based on hidden Markov model for real-time

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.

traffic sensing applications," in *Proc. IEEE 15th Int. IEEE Conf. Intelligent Transportation Systems*, 2012, pp. 776–781.
[23] F. R. Hapel, "The influence curve and its role in robust estimation," *J. Amer. Statist. Assoc.*, vol. 69, no. 346, pp. 383–393, 1974.
[24] M. Hashemi and H. A. Karimi, "A critical review of real-time map-matching algorithms: Current issues and future directions," *Comput. Environ. Urban Syst.*, vol. 48, pp. 153–165, 2014.
[25] S. K. Honey, W. B. Zavoli, K. A. Milnes, A. C. Phillips, M. S. White, Jr., and G. E. Loughmiller, Jr., "Vehicle navigational system and method," U.S. Patent 4 796 191, 1989.
[26] B. Hummel, "Map matching for vehicle guidance," in *Dynamic and Mobile GIS: Investigating Space and Time*. Boca Raton, FL: CRC Press, 2006.
[27] T. Hunter, P. Abbeel, and A. Bayen, "The path inference filter: Model-based low-latency map matching of probe vehicle data," *IEEE Trans. Intell. Transport. Syst.*, vol. 15, no. 2, pp. 507–529, 204.
[28] M. Jabbour, P. Bonnifait, and V. Cherfaoui, "Map-matching integrity using multihypothesis road-tracking," *J. Intell. Transport. Syst.*, vol. 12, no. 4, pp. 189–201, 2008.
[29] G. R. Jagadeesh, T. Srikanthan, and X. D. Zhang, "A map matching method for GPS based real-time vehicle location," *J. Navig.*, vol. 57, no. 3, pp. 429–440, 2004.
[30] K. Jones, L. Liu, and F. Alizadeh-Shabdiz, "Improving wireless positioning with look-ahead map-matching," in *Proc. IEEE 4th Annu. Int. Conf. Mobile and Ubiquitous Systems Networking and Services*, 2007, pp. 1–8.
[31] S. Kim and J.-H. Kim, "Adaptive fuzzy-network-based C-measure map-matching algorithm for car navigation system," *IEEE Trans. Ind. Electron.*, vol. 48, no. 2, pp. 432–441, 2001.
[32] K. John, E. Horvitz, and J. Letchner, "Map, matching with travel time constraints," Tech. Rep. SAE Technical Paper, 2007.
[33] M. Kubička, A. Cela, P. Moulin, H. Mounier, and S.-I. Niculescu, "Dataset for testing and training of map-matching algorithms," in *Proc. IEEE Intelligent Vehicles Symp.*, 2015, pp. 1088–1093.
[34] M. Kubička, A. Cela, P. Moulin, H. Mounier, and S. I. Niculescu. (2016, July). Dataset for testing and training map-matching methods. [Online]. http://dx.doi.org/10.5281/zenodo.57731
[35] M. Kubička, A. Cela, H. Mounier, and S.-I. Niculescu, "On designing robust real-time map-matching algorithms," in *Proc. IEEE 17th Int. Conf. Intelligent Transportation Systems*, Oct. 2014, pp. 464–470.
[36] L. Li, M. Quddus, and L. Zhao, "High accuracy tightly-coupled integrity monitoring algorithm for map-matching," *Transport. Res. Part C Emerging Technol.*, vol. 36, pp. 13–26, 2013.
[37] Y. Li, Q. Huang, M. Kerber, L. Zhang, and L. Guibas, "Large-scale joint map matching of GPS traces," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Advances Geographic Information Systems*, 2013, pp. 214–223.
[38] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Advances Geographic Information Systems*, 2009, pp. 352–361.
[39] F. Marchal, J. Hackney, and K. W. Axhausen, "Efficient map matching of large global positioning system data sets: Tests on speed-monitoring experiment in Zürich," *Transport. Res. Record J. Transport. Res. Board*, vol. 1935, no. 1, pp. 93–100, 2005.
[40] G. Nassreddine, F. Abdallah, and T. Denœux, "Map matching algorithm using belief function theory," in *Proc. IEEE 11th Int. Conf. Information Fusion*, 2008, pp. 1–8.
[41] G. Nassreddine, F. Abdallah, and T. Denœux, "Map matching algorithm using interval analysis and Dempster–Shafer theory," in *Proc. IEEE Intelligent Vehicles Symp.*, 2009, pp. 494–499.
[42] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Advances Geographic Information Systems*, 2009, pp. 336–343.
[43] V. Noronha and M. F. Goodchild, "Map accuracy and location expression in transportation: Reality and prospects," *Transport. Res. Part C Emerging Technol.*, vol. 8, no. 1, pp. 53–69, 2000.
[44] O. Pink and B. Hummel, "A statistical approach to map matching using road network geometry, topology and vehicular motion constraints," in *Proc. IEEE 11th Int. Conf. Intelligent Transportation Systems*, 2008, pp. 862–867.
[45] J.-S. Pyo, D.-H. Shin, and T.-K. Sung, "Development of a map matching method using the multiple hypothesis technique," in *Proc. Intell. Transport. Syst.*, 2001, pp. 23–27.
[46] M. A. Quddus, "High integrity map matching algorithms for advanced transport telematics applications," Ph.D. dissertation, Imperial College London, London, 2006.
[47] M. A. Quddus, R. B. Noland, and W. Y. Ochieng, "A high accuracy fuzzy logic based map matching algorithm for road transport," *J. Intell. Transport. Syst.*, vol. 10, no. 3, pp. 103–115, 2006.
[48] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transport. Res. Part C Emerging Technol.*, vol. 15, no. 5, pp. 312–328, 2007.
[49] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
[50] R. Raymond, T. Morimura, T. Osogami, and N. Hirosue, "Map matching with hidden Markov model on sampled road network," in *Proc. 21st Int. Conf. Pattern Recognition*, 2012, pp. 2242–2245.
[51] M. Ren, "Advanced map matching technologies and techniques for pedestrian/wheelchair navigation," Ph.D. dissertation, Univ. Pittsburgh, 2012.
[52] N. Schuessler and K. W. Axhausen, "Map-matching of GPS traces on high-resolution navigation networks using the multiple hypothesis technique (MHT)," Working paper 568, *Arbeitsberichte Verkehrsund Raumplanung*, 2009.
[53] S. Steiniger, M. Neun, A. Edwardes, and B. Lenz, "*Foundations of LBS*," CartouCHe: Cartography Swiss Higher Education, 2008.
[54] S. Syed and E. Cannon, "Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons," in *Proc. Institute of Navigation National Technical Meeting*, San Diego, CA, 2004, vol. 1, pp. 26–28.
[55] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proc. 8th USENIX Conf. Networked Systems Design and Implementation*, Berkeley, CA, 2011, pp. 267–280.
[56] R. Toledo-Moreo, D. Bétaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps," *IEEE Trans. Intell. Transport. Syst.*, vol. 11, no. 1, pp. 100–112, 2010.
[57] N. R. Velaga, "Development of a weight-based topological map-matching algorithm and an integrity method for location-based ITS services," Ph.D. dissertation, Loughborough Univ., 2010.
[58] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Survey Rev.*, vol. 23, no. 176, pp. 88–93, 1975.
[59] K. Virrantaus, J. Markkula, A. Garmash, V. Terziyan, J. V. Alainen, A. Katanosov, and H. Tirri, "Developing GIS-supported location-based services," in *Proc. 2nd Int. Conf. Web Information Systems Eng.*, 2001, vol. 2, pp. 66–75.
[60] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, no. 2, pp. 260–269, 1967.
[61] M. Weber, L. Liu, K. Jones, M. J. Covington, L. Nachman, and P. Pesti, "On map matching of wireless positioning data: A selective look-ahead approach," in *Proc. 18th ACM SIGSPATIAL Int. Conf. Advances Geographic Information Systems*, 2010, pp. 290–299.
[62] H. Wei, Y. Wang, G. Forman, and Y. Zhu, "Map matching by Fréchet distance and global weight optimization," Department of Computer Science and Engineering, Technical Paper, 2013.
[63] H. Wei, Y. Wang, G. Forman, and Y. Zhu, "Map matching: Comparison of approaches using sparse and noisy data," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Advances Geographic Information Systems*, 2013, pp. 444–447.
[64] H. Wei, Y. Wang, G. Forman, Y. Zhu, and H. Guan, "Fast viterbi map matching with tunable weight functions," in *Proc. ACM 20th Int. Conf. Advances Geographic Information Systems*, 2012, pp. 613–616.
[65] C. Wenk, R. Salas, and D. Pfoser, "Addressing the need for map-matching speed: Localizing global curve-matching algorithms," in *Proc. 18th Int. Conf. Scientific and Statistical Database Management*, 2006, pp. 379–388.
[66] C. E. White, D. Bernstein, and A. L. Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transportation Res. Part C Emerging Technol.*, vol. 8, no. 1, pp. 91–108, 2000.
[67] Z. Xiao, H. Wen, A. Markham, and T. Niki, "Lightweight map matching for indoor localisation using conditional random fields," in *Proc. 13th Int. Symp. Information Processing Sensor Networks*, 2014, pp. 131–142.
[68] D. Yang, B. Cai, and Y. Yuan, "An improved map-matching algorithm used in vehicle navigation system," in *Proc. IEEE Intelligent Transportation Systems*, 2003, vol. 2, pp. 1246–1250.
[69] J.-S. Yang, S.-P. Kang, and K.-S. Chon, "The map matching algorithm of GPS data with relatively long polling time intervals," *J. Eastern Asia Soc. Transport. Stud.*, vol. 6, pp. 2561–2573, 2005.
[70] X. Zhang, Q. Wang, and D. Wan, "The relationship among vehicle positioning performance, map quality, and sensitivities and feasibilities of map-matching algorithms," in *Proc. IEEE Intelligent Vehicles Symp.,ss* 2003, pp. 468–473.
[71] Y. Zheng and M. A. Quddus, "Weight-based shortest-path aided map-matching algorithm for low-frequency positioning data," in *Proc. Transportation Research Board 90th Annu. Meeting Number 11-2680*, 2011.

ITS

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on July 26,2021 at 05:47:04 UTC from IEEE Xplore. Restrictions apply.