A Survey on Map-Matching Algorithms

Preprint · October 2019

4 authors, including:

Pingfu Chao
Soochow University (PRC)
25 PUBLICATIONS   **55** CITATIONS

Yehong Xu
The Hong Kong University of Science and Technology
3 PUBLICATIONS   **14** CITATIONS

Wen Hua
The University of Queensland
75 PUBLICATIONS   **590** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project

Distributed entity matching at ECNU View project

A Survey on Map-Matching Algorithms

Pingfu Chao[1], Yehong Xu[1], Wen Hua[1], and Xiaofang Zhou[1]

School of Information Technology and Electrical Engineering,
The University of Queensland, Australia
{p.chao,yehong.xu,w.hua}@uq.edu.au, zxf@itee.uq.edu.au

Abstract. The map-matching is an essential preprocessing step for most of the trajectory-based applications. Although it has been an active topic for more than two decades and, driven by the emerging applications, is still under development. There is a lack of categorisation of existing solutions recently and analysis for future research directions. In this paper, we review the current status of the map-matching problem and survey the existing algorithms. We propose a new categorisation of the solutions according to their map-matching models and working scenarios. In addition, we experimentally compare three representative methods from different categories to reveal how matching model affects the performance. Besides, the experiments are conducted on multiple real datasets with different settings to demonstrate the influence of other factors in map-matching problem, like the trajectory quality, data compression and matching latency.

1 Introduction

Nowadays, the ubiquity of positioning devices enables the tracking of user/vehicle trajectories. However, due to the intrinsic inaccuracy of the positioning systems, a series of preprocessing steps are required to correct the trajectory errors. As one of the major preprocessing techniques, the map-matching algorithm finds the objects travel route by aligning its positioning data to the underlying road network. It is the prerequisite of various location-based applications, such as navigation, vehicle tracking, map update and traffic surveillance.

The map-matching problem has been studied for more than two decades. Despite hundreds of papers are proposed, to the best of our knowledge, only several works were conducted [4,8,14,19] surveying them. More importantly, even the most recent surveys [8] fail to categorise the existing methods comprehensively. They either classify them based on applications [8] that are not very distinctive to each other, or follow the previous categorisation [14] that is obsolete. Besides, various new techniques are introduced to the map-matching problem recently, including new models (weight-based [15], multiple hypothesis theory [16]), new tuning techniques (machine learning [12], information fusion [5,9]), new data types (DGPS, inertial sensor, semantic road network) and new research topics (lane-level, parallel). Hence, it is about time to conduct a new survey to summarise existing solutions and provide guidance to future research.

Note that the existing map-matching problem covers various scenarios, ranging from indoor to outdoor and from pedestrian, vehicle to multimodal. However,

to ensure a unified setting for survey and comparison, in this paper, we target the vehicle trajectory map-matching in an outdoor environment due to its popularity. We categorise the existing work from technical perspective. In addition, we discuss the main properties of the methods and future research directions according to the experiment results conducted on multiple matching algorithms. Overall, our contributions are listed as follows:

– We review the map-matching solutions proposed since the last comprehensive survey [14] and propose a new categorisation of the algorithms based on their methodology. Our proposed categorisation can better distinguish the existing methods from the technical perspective, which is beneficial for future study.
– We enumerate several map-matching challenges that are caused by low-quality trajectory data. The challenges are exemplified and explained concretely, which leads to future research directions.
– To further demonstrate the challenges, we implement three representative map-matching algorithms and conduct extensive experiments on datasets with different sampling rate, map density and compression level. Our claims about the relationship between data quality and map-matching quality are fully supported by the experiments.

The rest of the paper is organised as follows: In Section 2, we first formally define the map-matching problem and enumerate the existing surveys and their limitations. Then, we propose our new categorisation in Section 3. We further discuss the current challenges which are demonstrated through experiments in Section 4 and we draw conclusions in Section 5.

2   Preliminaries

2.1   Problem Definition

We first define the map-matching problem and relevant datasets, including trajectory (input), road network (input) and route (output):

Definition 1. *(Trajectory) A **trajectory** Tr is a sequence of chronologically ordered spatial points $Tr : p_1 \rightarrow p_2 \rightarrow ... \rightarrow p_n$ sampled from a continuously moving object. Each point p_i consists of a 2-dimensional coordinate $< x_i, y_i >$, a timestamp t_i, a speed spd_i (optional) and a heading θ_i (optional). i.e.: $p_i =< x_i, y_i, t_i, spd_i, \theta_i >$.*

Definition 2. *(Road Network) A **road network** (also known as map) is a directed graph $G = (V, E)$, in which a vertex $v = (x, y) \in V$ represents an intersection or a road end, and an edge $e = (s, e, l)$ is a directed road starting from vertices s to e with a polyline l represented by a sequence of spatial points.*

Definition 3. *(Route) A route R represents a sequence of connected edges, i.e. $R : e_1 \rightarrow e_2 \rightarrow ... \rightarrow e_n$, where $e_i \in G.E (1 \leq i \leq n)$ and $e_k.e = e_{k+1}.s$.*

Definition 4. *(Map-Matching) Given a road network $G(V, E)$ and a trajectory Tr, the map-matching find a route $\mathcal{MR}_G(Tr)$ that represents the sequence of roads travelled by the trajectory.*

For simplicity, we omit the subscript G and use $\mathcal{MR}(Tr)$ instead to represent the matching result as different trajectories are usually map-matched on the same map. In general, the map-matching route is expected to be continuous as it represents the vehicle's travel history. However, it is quite often that $\mathcal{MR}(Tr)$ contains disconnected edges due to incorrect map-matching, which will be discussed in Section 4.

2.2   Related Work

Intuitively, since the vehicle usually runs on the roads, a fully accurate trajectory sampled from a vehicle should always lie on the map. Therefore, apart from some unexpected map errors, which happens less frequently and is addressed by map update process [2], the difficulty of map-matching problem solely depends on the quality of the input trajectories. As studied in many papers, the quality issues in trajectories are pervasive, which mainly caused by inaccurate measurement and low sampling rate. In terms of the *measurement error*, due to the unstable connection between GPS device and satellites, the location of GPS samples usually deviate from its actual position by a random distance. Meanwhile, the *sampling error* is mainly caused by lowering the sampling frequency.

To deal with the quality issues, the map-matching problem has been studied for more than two decades. In terms of the working scenarios and applications, the current map-matching solutions can be classified into online mode and offline mode. In online map-matching, the vehicle positions are sampled continuously and are processed in a streaming fashion, which means each time the map-matching is only performed on the current sample with a limited number of preceding or succeeding samples [3,21] as reference. The process is usually simple and fast for interactive performance. In contrary, the offline map-matching is performed after the entire trajectory is obtained, so it aims for optimal matching route with less constraint on processing time.

From the methodology perspective, Quddus et al. [14] first conducted a comprehensive review of the map-matching algorithms proposed before 2007. The paper classified the methods into four categories, namely *geometric*, *topology*, *probabilistic* and *advanced*. The *geometric* methods only focus on the distance between trajectory elements and the road network, while the *topology* methods take into consideration the connectivity and shape similarity. The *probabilistic* methods try to model the uncertainty of trajectory, including the measurement error and the unknown travel path between two samples, and they aim to find a path that has the highest probability to generate the given trajectory. The *advanced* category contains methods that are based on some advanced models, like Kalman Filter, particle filter and fuzzy logic. This categorisation shows the evolution of map-matching research, which starts from simple, fast but inaccurate geometric-based methods to more complicated but accurate probability/advanced solutions. It is by far the most comprehensive survey of this field.

However, after more than ten years' development, most of the methods mentioned in the paper has been outperformed by their new successors and the previous categorisation also requires a revisit. Several surveys proposed afterwards reviewed the methods in certain perspectives. Hashemi et al. [4] targeted at the online map-matching scenario. Kubička et al. discussed the map-matching problem based on the applications [8], namely *navigation*, *tracking* and *mapping*. Other categorisations also appear recently (*incremental max-weight*, *global max-weight* and *global geometry* [19]) which shows that there is still no consensus on how to classify the algorithms technically. However, all of the existing categorisations inherit the same idea from Quddus' survey [14] with minor variations, which fail to categorise the recent methods for multiple reasons, explained in Section 3.

3   Survey of Map-Matching Algorithm

According to our study, previous categorizations fail to classify the current solutions due to three main reasons: (1) Categories for some primary methods, such as *geometric* category [14], are no longer the focus due to their weak performance. (2) Application-based classification [4,8] cannot fully distinguish the methods. Many of the map-matching algorithms, like the Hidden Markov Model (HMM) and Multiple Hypothesis Technique (MHT), apply to both online and offline scenarios for different applications. (3) Classifying algorithms by embedded mathematical tools are not feasible since many recent algorithms employ multiple mathematical tools. Furthermore, the same tool implemented in different algorithms may be used for different purposes, for example, an extended Kalman filter can be used to either estimate biases in GPS or fuse measurements from different sources [9].

Therefore, we establish a new classification that classifies the map-matching algorithms by their core matching model, which is employed to coordinate their techniques to finally achieve map-matching. In a map-matching algorithm, the map-matching model is the overall framework or matching principle for the map-matching process. A model usually consists of a set of computation components, like the calculation of distance, transition and user behaviour modelling, and a workflow connecting them. Those components are fixed while their definition and implementation vary among different methods. Existing map-matching models can be categorised into four classes: *similarity model*, *state-transition model*, *candidate-evolving model* and *scoring model*.

3.1   Similarity Model

The similarity model refers to a general approach that returns the vertices/edges that is *closest* to the trajectory geometrically and/or topologically. Intuitively, since a vehicle's movement always follows the topology of the underlying road network and the vehicle can never leap from one segment to another, the trajectory should also similar to those of the true path on the map. Therefore, the main focus in this category is how to define the *closeness*.

Distance-based Most of the earliest point-to-curve and curve-to-curve matching algorithms [14] follow this idea. Specifically, the point-to-curve solution projects each trajectory point to the geometric-closest edge, whereas the curve-to-curve matching algorithms project each trajectory segment to the closest edge where the *closeness* is defined by various similarity metrics. Fréchet distance is the most commonly-used distance function [18] since it considers the monotonicity and continuity of the curves. However, it is sensitive to trajectory measurement errors since its value can be dominated by the outliers. As an alternative, Longest Common Subsequence (LCSS) [23] divide a trajectory into multiple segments and find the shortest path on the map for each pair of start and end points of a trajectory segment. The shortest paths are then concatenated and form the final path while their corresponding LCSS scores are summed as the final score. Then, the path whose LCSS score is higher than a predefined threshold is regarded as the final matching result.

Pattern-based The pattern-based algorithms utilise the historical map-matched data to answer new map-matching queries by finding similar travel patterns [22]. The assumption is that people tend to travel on the same paths given a pair of origin and destination points. Therefore, by referring to the historical trajectories that are similar to the query trajectory, its candidate paths can be obtained without worrying about the sparseness of trajectory samples. Specifically, a historical trajectory or a trajectory obtained by concatenating multiple historical trajectories will be referred to if each point of this trajectory is in the safe region around the query trajectory. The algorithm finally uses a scoring function to decide the optimal route. However, due to the sparsity and disparity of historical data, the query trajectory may not be fully covered by historical trajectories especially in some rarely travelled regions, which leads to a direct matching process.

3.2 State-Transition Model

The state-transition models build a weighted topological graph which contains all possible routes the vehicle might travel. In this graph, the vertices represent the possible *states* the vehicle may be located at a particular moment, while the edges represent the *transitions* between states at different timestamps. Different from the road network, the weight of a graph element represents the possibility of a state or a transition, and the best matching results comes from the optimal path in the graph globally. There are three major ways of building the graph and solving the optimal path problem, namely Hidden Markov model (HMM), Conditional Random Field (CRF) and the Weighted Graph Technique (WGT).

Hidden Markov model HMM is one of the most widely used map-matching models as it simulates the road network topology meanwhile considers the reasonability of a path. HMM focuses on the case when states in the Markov chain

are unobservable (hidden) but can be estimated according to the given observations associated with them. This model fits in the map-matching process naturally. Each trajectory sample is regarded as the observation, while the vehicle actual location on the road, which is unknown, is the hidden states. In fact, due to the trajectory measurement error, all the roads near the observation can potentially be the actual vehicle location (state), each of which with a probability (emission probability). As the trajectory travels continuously, the transition between two consecutive timestamps is concluded by the travel possibility (transition probability) between their candidate states. Therefore, the objective is to find an optimal path which connects one candidate in every timestamp. The final path is obtained by the Viterbi algorithm which utilises the idea of dynamic programming. The major difference between various HMM-based algorithms is their definition of emission probability and transition probability. Unlike the emission probability, which is defined identically in most papers, the definition of the transition probability varies since the travel preference can be affected by plenty of factors. Some works [11] prefers a candidate pair whose distance is similar to the distance between the observation pair, while others consider velocity changes [3], turn restriction [12], closeness to the shortest path, the heading mismatch and travel penalty on U-turns, tunnels and bridges. Besides, HMM is also applied to online scenario [3]. However, to build a reasonable Markov chain, online HMM-based algorithms usually suffer from latency problems, which means a point is matched after a certain delay.

Conditional random field CRF is utilized in many areas as an alternative to HMM to avoid the selection bias problem [6]. As both CRF and HMM are statistic models, the major difference is that CRF models interactions among observations while HMM models only model the relation between an observation with the state at the same stage and its closest predecessor. Hunter et al. [6] proposed a CRF-based map-matching algorithm that can be applied to both online and offline situations with high accuracy. Its overall approach is similar to HMM-based algorithms but with different transition probability which considers the maximum speed limit and the driving patterns of drivers. However, the problem shared by both HMM and CRF is the lack of a recovery strategy for the match deviation. Since once a path is confirmed, it will be contained by all future candidate paths, which hurts the online scenarios especially.

Weighted graph technique WGT refers to a model that infers the matching path from a weighted candidate graph, where the nodes are candidate road points of location measurements and edges are only formed between two nodes corresponding to two consecutive samples. In most WGT-based algorithms, candidate points are the closest points on road segments in a radius of measurements [5,10], which is similar to HMM. The process of the WGT can be summarized as three steps: (1) Initializing the candidate graph. (2) Weighting edges in the graph using a scoring function. (3) Inferring a path based on the weighted graph.

Algorithms fall in this category mainly differs from each other in weighting functions. Lou et al. [10] firstly propose the WGT. It weights an edge simply based on a spatial cost and a temporal cost, where the spatial cost is modelled on the distance between candidate c_i to its observed position p_i and the shortest length between c_i and c_{i+1} whereas the temporal cost is modelled on the velocity reasonability. Based on Lou's design, the following work further considers mutual influences between neighbouring nodes, road connectivity, travel time reasonability [5] and other road features (traffic lights, left turns, etc.).

3.3   Candidate-Evolving Model

Candidate-evolving model refers to a model which holds a set of candidates (also known as particles or hypotheses) during map-matching. The candidate set is initiated based on the first trajectory sample and keeps evolving by adding new candidates propagated from old ones close to the latest measurements while pruning irrelevant ones. Interpreting a candidate as a vote, by maintaining the candidate set, the algorithms are able to find a segment with the most votes, thereby, determining the matching path. Compare to the state-transition model, the candidate-evolving model is more robust to the off-track matching issue since the current matching is influenced not only by a previously defined solution, but also by other candidates. The particle filter (PF) and the Multiple Hypothesis Technique (MHT) are two representative solutions.

Particle filter PF is a state estimation technique that combines Monte Carlo sampling methods with Bayesian Inference. This technique has been utilized to support map-matching by the way of sensor fusion and measurement correction [17], while it is also applicable to directly address map-matching problem [1]. The general idea of the PF model is to recursively estimate the Probability Density Function (PDF) of the road network section around the observation as time advances. Here, the PDF is approximated by N discrete particles, each particle maintains a weight representing how consistent it is to the location observation. The process of a PF can be summarized as follows: Initially, N particles are sampled with the same weight representing different locations in the local road network. The weight of each particle keeps getting updated as new observations are received. Then the PDF for the road network section around the new observations is calculated and the area with the highest probability is determined as the matched region. A resampling stage starts afterwards, where a new set of particles are derived based on the current set. The particles with higher weights are more likely to propagate according to moving status to feed particles for the next cycle, while those with low weights are likely to die out.

Multiple hypothesis technique Similar to PF, the MHT also tries to maintain a list of candidate road matches for the initial trajectory point and the list is expected to be as large as possible to ensure correct result coverage. However, different from the PF which iterate through all possibilities over time, the MHT

is a much simpler model that inherits the idea of maintaining hypotheses but manages to reduce computation during the process. An MHT evaluates each candidate road edge (or point) based on a scoring function instead of trying to approximate the complicated PDF for the neighbour map area. Thereby, the computation cost of the MHT is dramatically reduced. According to the intuition, the MHT can be easily adopted in online scenario [16]. Moreover, since it possesses all the possibility of previous hypotheses, Taguchi et al. [16] propose a prediction model which extends the hypotheses to further predict the future route, which can achieve better online map-matching accuracy without introducing latency.

3.4 Scoring Model

Naïve weighting A group of algorithms [13,15] apply the weight without using a particular model. Instead, they simply assign a group of candidates to each trajectory segment (or location observation) and find a road edge from each group that maximizes the predefined scoring function. The found segment in every timestamp is either returned if applied to the online scenario or waited to be joint with other matched segments if applied in the offline scenario. Most recent work in this category [15] achieves a lane-level map-matching performance. The algorithm first identifies lanes in each road by utilising the road width information in the map and partition them into grids accordingly. The algorithm then finds candidate lane grids around the observed location and scores these grids at each timestamp. The grid results in the maximum score are then returned. The scoring function is a linear combination of four features, i.e. the proximity between the grid and trajectory sample, the estimated location of the vehicle at the next time stage, the reachability from the grid and the intention of a turn. These features are modelled individually, their scores can be obtained from the corresponding models in every timestamp. In addition, feature scores are weighted differently in the scoring function whose coefficients are computed by a training process before map-matching starts.

4 Challenges and Evaluations

Despite various of map-matching models are proposed to deal with trajectory quality issues, the current solutions still fail to achieve decent matching quality in all scenarios. Therefore, in this section, we will discuss several major challenges caused by data quality issues that are affecting the map-matching results. We will demonstrate them both visually and experimentally to exemplify their significance.

4.1 Experimental Settings

As listed in Table 1, we use four datasets for our experiments. The *Global* [7] dataset is a public dataset for map-matching evaluation. It contains 100 GPS trajectories sampled from 100 different areas all over the world, each of which is

provided with a dedicate underlying map. Besides, we extract three sub-areas, namely *Beijing-U*, *Beijing-R* and *Beijing-M*, from a commercial dataset which contains taxi trajectories in Beijing. The reason for choosing these four datasets is their diversity in terms of trajectory quality and map density. The *Global* dataset has the best trajectory accuracy and its maps are also very sparse. The *Beijing-U* and *Beijing-R* represent two maps extracted from urban and rural areas, respectively. They have roughly the same size but different map density ($27.3vs13.9$), so they can be used to evaluate the influence of map density to map-matching results. *Beijing-M* is a larger map area with more trajectories for large-scale performance test.

Table 1. Summary of experiment datasets

Name	Input Trajectory			Road Network			
	Trajectory Count	*Trajectory Point Count*	*Sampling rate(sec)*	*# of vertices + mini nodes*	*# of edges*	*Map Size(km^2)*	*Map Density (km/km^2)*
Global	100	N/A	1	N/A	N/A	N/A	N/A
Beijing-U	7,905	247,544	11.0	7,672	4,484	9.9	27.3
Beijing-R	3,106	119,612	8.6	3,927	1,326	9.9	13.9
Beijing-M	73,072	3,285,934	10.3	41,353	22,580	57.0	24.2

Our experiments are performed on a single server with two Intel(R) Xeon(R) CPU E5-2630 with 10 cores/20 threads at 2.2GHz each, 378GB memory and Ubuntu 16.04. Both the route matching result $\mathcal{MR}(Tr)$ and the corresponding ground-truth are regarded as sets of road edges and are evaluated by F-measure, which is commonly used in map-matching evaluation [15,19]. The candidate map-matching algorithms used in the experiments include the most popular offline HMM map-matching [11], the most-recent offline WGT algorithm [20] and an online Scoring method [13].

4.2 Data Quality Challenges

According to our observations from the experiments, the current data quality issues affect the map-matching in three major ways: the unnecessary detours, the matching breaks and the matching uncertainty.

Unnecessary detour As an example shown in Fig. 1a, the matching result sometimes may contain unnecessary detours, which happens more frequently when the trajectory sampling rate is very high. In most scenarios, the detour is caused by two consecutive trajectory samples being too close to each other so that the succeeding point happens to be matched to the upper stream of its preceding point. Therefore, the shortest path between these two points has to go through a long detour. To avoid such issue, the measurement error should be considered when finding the shortest path, which means a certain degree of backtrace should be allowable. Alternatively, instead of simply project trajectory samples to the candidate roads to find candidate points, the actual matching point should follow a distribution, according to the trajectory measurement error, along the candidate road.
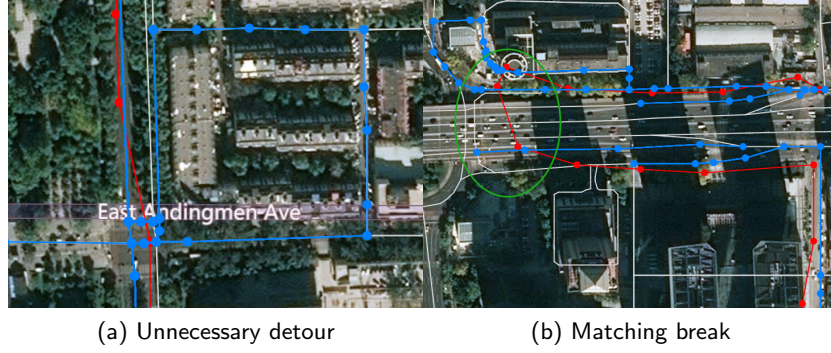
(a) Unnecessary detour   (b) Matching break

Fig. 1. Example of map-matching challenges

In general, as depicted in Fig. 2a, the detour problem strongly affects the matching quality when the sampling rate is high. The result shows that it is not always the case that a higher sampling rate leads to higher matching quality especially when the measurement error becomes the major problem. Therefore, a better way of modelling the measurement error is still required.
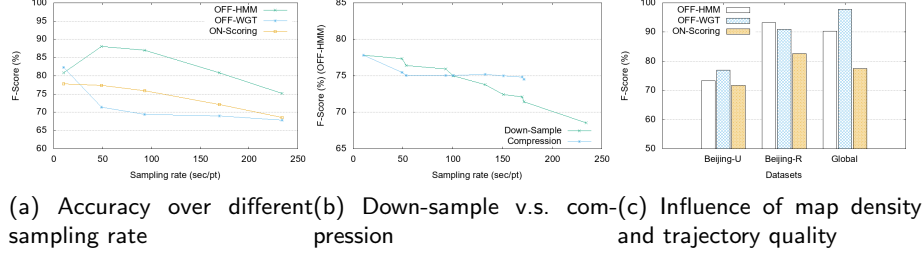

(a) Accuracy over different (b) Down-sample v.s. com- (c) Influence of map density
sampling rate   pression   and trajectory quality

Fig. 2. Experimental results

Matching break The matching break is a common problem in map-matching, which is mainly caused by trajectory outliers. This happens more frequently in the state-transition matching model when the correct state falls out of the candidate range of the outlier. In this case, the states of two consecutive observations may be unreachable, leading to disconnected matching route, as shown in the green circled area in Fig. 1b. Currently, most of the solutions [11] try to overcome this problem by identifying and removing the outliers to remedy the broken route. In Fig. 2b, we apply online scoring method on *Beijing-M* with random down-sample and trajectory compression (Douglas-Peucker algorithm), respectively. The result shows that simple trajectory compression fails to prune outliers as they are usually preserved as outstanding point, which means more preprocessing step is required to remove such outliers. However, considering the detour problem in high sampling rate data, the trajectory compression achieves better performance compared with simply down-sample the trajectory as it better preserve the shape of the trajectory, which is still beneficial.

Matching uncertainty Although the main goal of map-matching algorithms is to reduce the uncertainty of trajectory, the matching uncertainty varies in different scenarios. One of the main factor, which is not mentioned by any of previous work, is the map density. Intuitively, the map-matching of trajectory is much harder when the map area is full of roads compared with an emptier region. As shown in Fig. 2c, the map density can significantly affect the matching quality as the *Beijing-U* has much worse performance than *Beijing-R* given both of them have a similar trajectory quality. On the other hand, the trajectory quality also plays an important role since the performance on *Global* is better than on *Beijing-U* with similar map density. Therefore, achieving decent performance on dense map area is still a challenging task for future map-matching research.

5   Conclusion

In this paper, we conduct a comprehensive survey of the map-matching problem. We reveal the inability of all previous surveys in classifying new map-matching solutions. On top of that, we propose a new categorisation of existing methods from the technical perspective, which consists of similarity model, state-transition model, candidate-evolving model and scoring model. In addition, we list three major challenges (unnecessary detour, matching break and matching uncertainty) that the current map-matching algorithms are facing. To exemplify and demonstrate their influence on the current map-matching algorithms, we conduct extensive experiments over multiple datasets and map-matching algorithms. Overall, this paper concludes the current state of the map-matching problem and provides guidance to future research directions.

References

1. Bonnifait, P., Laneurit, J., Fouque, C., Dherbomez, G.: Multi-hypothesis map-matching using particle filtering. In: 16th World Congress for ITS Systems and Services. pp. 1–8 (2009)
2. Chao, P., Hua, W., Zhou, X.: Trajectories know where map is wrong: an iterative framework for map-trajectory co-optimisation. World Wide Web pp. 1–27 (2019)
3. Goh, C.Y., Dauwels, J., Mitrovic, N., Asif, M.T., Oran, A., Jaillet, P.: Online map-matching based on hidden markov model for real-time traffic sensing applications. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems. pp. 776–781. IEEE (2012)
4. Hashemi, M., Karimi, H.A.: A critical review of real-time map-matching algorithms: Current issues and future directions. Computers, Environment and Urban Systems **48**, 153–165 (2014)
5. Hu, G., Shao, J., Liu, F., Wang, Y., Shen, H.T.: If-matching: Towards accurate map-matching with information fusion. IEEE Transactions on Knowledge and Data Engineering **29**(1), 114–127 (2017)
6. Hunter, T., Abbeel, P., Bayen, A.: The path inference filter: model-based low-latency map matching of probe vehicle data. IEEE Transactions on Intelligent Transportation Systems **15**(2), 507–529 (2014)

7. Kubička, M., Cela, A., Moulin, P., Mounier, H., Niculescu, S.I.: Dataset for testing and training of map-matching algorithms. In: Intelligent Vehicles Symposium (IV), 2015 IEEE. pp. 1088–1093. IEEE (2015)
8. Kubicka, M., Cela, A., Mounier, H., Niculescu, S.I.: Comparative study and application-oriented classification of vehicular map-matching methods. IEEE Intelligent Transportation Systems Magazine **10**(2), 150–166 (2018)
9. Li, L., Quddus, M., Zhao, L.: High accuracy tightly-coupled integrity monitoring algorithm for map-matching. Transportation Research Part C: Emerging Technologies **36**, 13–26 (2013)
10. Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., Huang, Y.: Map-matching for low-sampling-rate gps trajectories. In: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems. pp. 352–361. ACM (2009)
11. Newson, P., Krumm, J.: Hidden markov map matching through noise and sparseness. In: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems. pp. 336–343. ACM (2009)
12. Osogami, T., Raymond, R.: Map matching with inverse reinforcement learning. In: Twenty-Third International Joint Conference on Artificial Intelligence (2013)
13. Quddus, M., Washington, S.: Shortest path and vehicle trajectory aided map-matching for low frequency gps data. Transportation Research Part C: Emerging Technologies **55**, 328–339 (2015)
14. Quddus, M.A., Ochieng, W.Y., Noland, R.B.: Current map-matching algorithms for transport applications: State-of-the art and future research directions. Transportation research part c: Emerging technologies **15**(5), 312–328 (2007)
15. Sharath, M., Velaga, N.R., Quddus, M.A.: A dynamic two-dimensional (d2d) weight-based map-matching algorithm. Transportation Research Part C: Emerging Technologies **98**, 409–432 (2019)
16. Taguchi, S., Koide, S., Yoshimura, T.: Online map matching with route prediction. IEEE Transactions on Intelligent Transportation Systems **20**(1), 338–347 (2018)
17. Wang, X., Ni, W.: An improved particle filter and its application to an ins/gps integrated navigation system in a serious noisy scenario. Measurement Science and Technology **27**(9), 095005 (2016)
18. Wei, H., Wang, Y., Forman, G., Zhu, Y.: Map matching by fréchet distance and global weight optimization. Technical Paper, Departement of Computer Science and Engineering p. 19 (2013)
19. Wei, H., Wang, Y., Forman, G., Zhu, Y.: Map matching: comparison of approaches using sparse and noisy data. In: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. pp. 444–447. ACM (2013)
20. Yang, C., Gidofalvi, G.: Fast map matching, an algorithm integrating hidden markov model with precomputation. International Journal of Geographical Information Science **32**(3), 547–570 (2018)
21. Yin, Y., Shah, R.R., Wang, G., Zimmermann, R.: Feature-based map matching for low-sampling-rate gps trajectories. ACM Transactions on Spatial Algorithms and Systems (TSAS) **4**(2), 4 (2018)
22. Zheng, K., Zheng, Y., Xie, X., Zhou, X.: Reducing uncertainty of low-sampling-rate trajectories. In: 2012 IEEE 28th International Conference on Data Engineering. pp. 1144–1155. IEEE (2012)
23. Zhu, L., Holden, J.R., Gonder, J.D.: Trajectory segmentation map-matching approach for large-scale, high-resolution gps data. Transportation Research Record **2645**(1), 67–75 (2017)