

PAPER • OPEN ACCESS

A Self-adjusting Online Map Matching Method

To cite this article: Weijia Lv *et al* 2021 *J. Phys.: Conf. Ser.* **2006** 012035

View the [article online](#) for updates and enhancements.

You may also like

- [Precise 3D foot-mounted indoor localization system using commercial sensors and map matching approach](#)
Lin Qi, Yue Yu, Yu Liu *et al.*
- [Fast approximate delivery of fluence maps for IMRT and VMAT](#)
Marleen Balvert and David Craft
- [Variance optimization of UWB observation based on map matching for PPP/INS/UWB tightly coupled positioning](#)
Zengke Li, Long Zhao, Yifan Wang *et al.*

A promotional banner for 'Free the Science Week 2023' with a dark blue background and a futuristic circular interface. A hand is shown interacting with the interface. The text 'Free the Science Week 2023' is in light blue, followed by 'April 2-9' in white. Below this, 'Accelerating discovery through' is in white and 'open access!' is in light blue. At the bottom left is the ECS logo and the website 'www.ecsdl.org'. At the bottom right is a blue button with the text 'Discover more!'.

Free the Science Week 2023 April 2-9

Accelerating discovery through
open access!

 www.ecsdl.org [Discover more!](#)

A Self-adjusting Online Map Matching Method

Weijia Lv^{1*}, Yi Chen¹, Pengfei Shang¹, Fang Zheng¹ and Jielong Wang¹

¹ College of Surveying and Geo-Informatics, Tongji University, Shanghai, 200092, China

*Corresponding author's e-mail: weijialv@tongji.edu.cn

Abstract. With the rapid development of intelligent transportation systems (ITS) and location-based services (LBS), it is more important to match the original trajectory sequence generated by users/vehicles to the actual road network. Most of the existing online map matching algorithms are based on the idea of local processing, or require richer data input and more mathematical models to ensure matching accuracy. This paper presents a simple and effective map matching method, called self-adjusting online map matching (AOMM). The algorithm is developed based on hidden Markov model (HMM). Considering the topological and geometric properties of the road network, the emission probability and transition probability calculation formulas of HMM are defined. And three adjustment strategies are provided to deal with trajectory noise points, dense trajectory points, and offset trajectory points. The algorithm only needs latitude and longitude information of trajectory points, and can match point by point. Experimental results on open trajectory data show that the algorithm has high matching accuracy and low output delay, and can meet the requirements of general online map matching tasks.

1. Introduction

Nowadays, with the development and popularization of the navigation systems, we can easily obtain numerous location data from various positioning devices. However, due to the positioning error and sampling error of the sensor, there is a deviation between the position obtained directly from the devices and the actual position of the users/vehicles[1]. This requires map matching to associate the position of devices with the road network of the electronic map so that users/vehicles can be correctly displayed on the road. Map matching is a key step in location-based services and an important prerequisite for better vehicle navigation[2], route planning/recommendation[3-4], travel/route prediction[5-6], etc. The existing map matching algorithms can be divided into geometric matching algorithm[7-9], topological relationship algorithm[10-11], probability statistics algorithm[12-13], and advanced matching algorithm[14-16] according to the information involved in the input data. According to the real-time performance of the calculation, map matching algorithms can be divided into online and offline algorithms. The offline algorithm is applied after the data record is completed, and all trajectory points can be considered for processing. It can accept the slow calculation speed, which is conducive to the improvement of matching accuracy. Therefore, some global algorithms[17-19] are more suitable for offline map matching tasks. The online algorithm is to complete the real-time matching of trajectory points in the process of data recording, which is characterized by fast calculation speed and can adapt to the online map matching task with a high sampling rate. Thence, some local algorithms[20-22] are more suitable. However, with the decrease of sampling rate, the matching accuracy of the online algorithms are usually difficult to be guaranteed. This requires the map matching algorithm to achieve a balance between accuracy and performance.



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Many application scenarios in real life (such as traffic supervision and route navigation) put forward high requirements for the real-time performance of map matching. Some local algorithms use the fixed sliding window strategy to solve the online matching problem. A larger window size leads to higher accuracy[17,23], but at the same time increases output delay. The algorithm that adopts the fixed-depth recursive look-ahead strategy is to evaluate future path alternatives by delaying the resolution of each trajectory point by a fixed number of steps[24]. Although these two methods are simple to be implemented, they may lead to suboptimal solutions and long output delay. When real-time applications require output in a short time window, such algorithms are obviously not expected. Some online map matching methods with variable window strategies usually require other information such as speed and direction as supplements in addition to the necessary latitude and longitude information[25]. To a certain extent, it is beneficial to improve the matching accuracy, but it also increases the complexity of the algorithm at the same time. However, some positioning devices only provide necessary data (such as timestamp, longitude, and latitude) and lack speed and direction information, which is difficult to apply the algorithm directly.

Considering the requirements of low output delay and high matching accuracy in online map matching tasks, this paper develops a self-adjusting online map matching algorithm based on hidden Markov model[26]. We define the calculation equations of emission probability and transition probability in HMM. Three adjustment strategies are proposed to ensure more stable and efficient execution of the algorithm. Matching experiments under different sampling rates are carried out using open data sets to evaluate the algorithm.

This paper is organized as follows. Section 2 describes how our method works for map matching. Our experiments are explained in Section 3. Finally, Section 4 summarizes our contributions and discusses the space for future works.

2. AOMM map matching algorithm

The algorithm in this paper is developed based on HMM. For trajectory points at each time, a set of candidate segments is firstly determined, each of which is represented as a hidden state in the Markov chain and has an emission probability that is the likelihood of observing the trajectory point. In general, if the trajectory point is close to a certain segment, we tend to specify a higher emission probability value for this segment. Then, the weight for each pair of adjacent vertices in a Markov chain is calculated, namely the transition probability. Finally, the maximum likelihood path with the highest joint emission probability and transition probability is found on the Markov chain[27]. This problem is usually solved by the Viterbi algorithm[28]. In fact, a dynamic programming algorithm is used to solve the hidden Markov model prediction problem, that is, dynamic programming is used to quickly find the optimal path to maximize the product of emission probability and transition probability in road networks. The basic framework of HMM for map matching is shown in Figure 1.

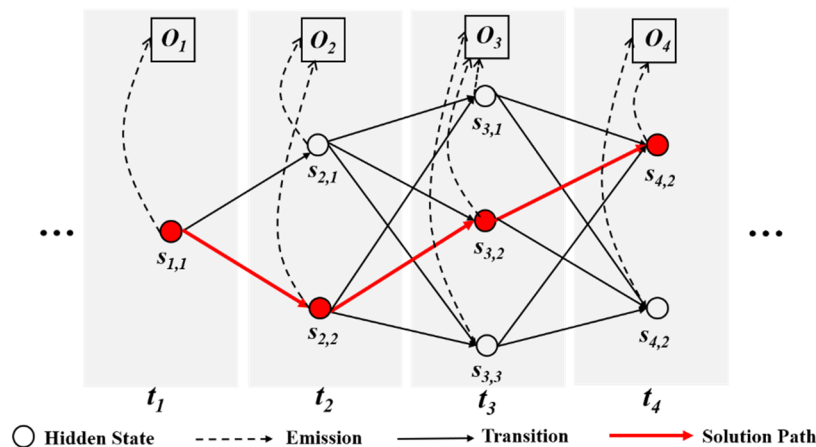


Figure 1. Basic framework of HMM for map-matching

2.1. Basic flow of the algorithm

- For each input trajectory point, search all candidate segments within a 50m radius around it. The reason for setting the threshold is to give up the candidate segments with low emission probability and avoid slowing down the algorithm execution due to too many candidate segments.
- The emission probability of each candidate segment is calculated. If the current trajectory point is not the starting point, the transition probability is calculated by using the candidate point of the moment and the candidate point of the previous moment.
- The Viterbi algorithm is used to solve the optimal matching path between the current trajectory point and the previous trajectory point, and the trajectory point is projected to the corresponding segment to obtain the optimal matching point.
- The optimal matching segment and the optimal matching point of the current trajectory point are stored as the candidate segment and candidate point for the next stage of matching.
- The adjustment strategies are applied to the matching process of each trajectory point to update the Markov chain and give a new solution, if available.
- Repeat the above process for the next trajectory point. When reaching the last trajectory point, the algorithm terminates.

2.2. Emission probability

As shown in Figure 2(a), the trajectory point O_t at time t has three candidate segments within the radius of 50m. Inspired by [29], the projection distances from O_t to three candidate segments are calculated respectively (if the projection point is on the extension line of the section, the distance from the trajectory point to the nearest point of the segment is calculated, such as $d_{t,1}$, $d_{t,3}$). The calculation formula of emission probability is as follows:

$$p(o_t | s_{t,k}) = \frac{1/d_{t,k}}{\sum 1/d_{t,k}} \quad (1)$$

2.3. Transition probability

As shown in Figure 2(b), calculate the distance $z(o_{t-1}, o_t)$ between two adjacent trajectory points, use A^* algorithm[30] to find the shortest path between projection points, and calculate the path length $r(s_{t-1,1}, s_{t,1})$, $r(s_{t-1,1}, s_{t,2})$. The transition probability is calculated by:

$$p(s_{t-1,j}, s_{t,k}) = \frac{1/|z(o_{t-1}, o_t) - r(s_{t-1,j}, s_{t,k})|}{\sum 1/|z(o_{t-1}, o_t) - r(s_{t-1,j}, s_{t,k})|} \quad (2)$$

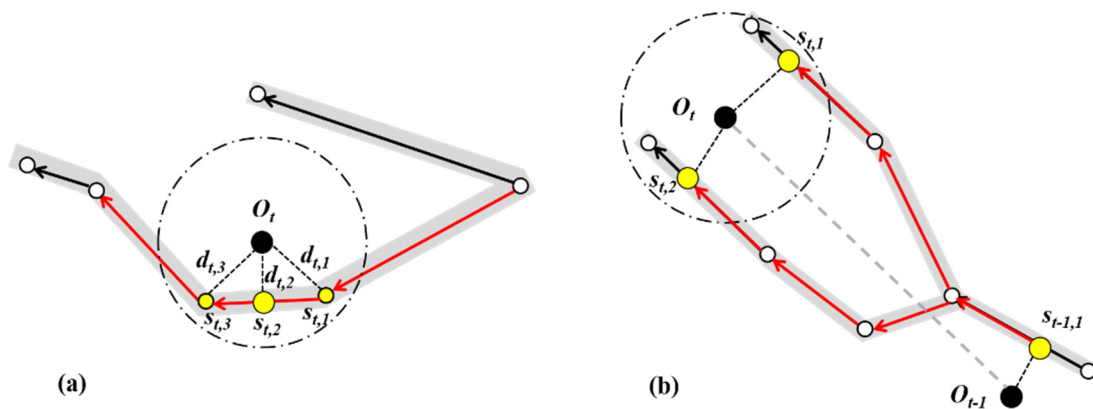


Figure 2. Examples of calculation of emission probability and transition probability

2.4. Adjustment strategy

2.4.1. Strategy I

When no candidate segment is found in the range of 50m radius around a certain trajectory point, our adjustment strategy is to skip the trajectory point and continue to match the next trajectory point.

2.4.2. Strategy II

In high sampling rates, moving objects are usually very slow (even $0m/s$) when traffic jams and signal lights wait, and the distance between adjacent trajectory points will become very small. If these trajectory points are still matched, it may not be obvious to visualize the matching results on the map. In addition, if the moving object does not move at this time, due to the trajectory point drift caused by the error of the positioning device, continuous matching of such trajectory points is little significant and easy to cause error matching. Therefore, the strategy designed for this kind of situation is to set the distance threshold d_{min} (This paper sets $d_{min}=3m$.) between the trajectory point at the current moment and the trajectory point at the previous moment. When $z(o_{t-1}, o_t) < d_{min}$, the trajectory point O_t is ignored, and the matching of the next trajectory point is continued.

2.4.3. Strategy III

The online algorithm can only use the historical trajectory point information to match before obtaining the position of the next trajectory point. When the previous moment of the trajectory point error matching (such as O_{t-1} in Figure 3(a)), after the next trajectory point input, if continue to use the matching result of O_{t-1} , we cannot match the current moment of the trajectory point O_t (there is no effective connected path in the road network). The strategy adopted at this time is to look forward to the trajectory point at time $t-2$, and continue to match O_t with the matching result of O_{t-2} . If the matching is still unable to be completed (Figure 3(b)), then continue to look forward to the trajectory points at time $t-3$, $t-4$..., and repeat the above process until the matching is successful (Figure 3(c)). Although this process can cause output delay to a certain extent, it can reduce the occurrence of mismatch, correct the route in time, and ensure the stability and matching accuracy of the algorithm. To meet the short-time output requirements under high sampling rate conditions, the maximum number of forwarding search trajectory points is generally set to 3.

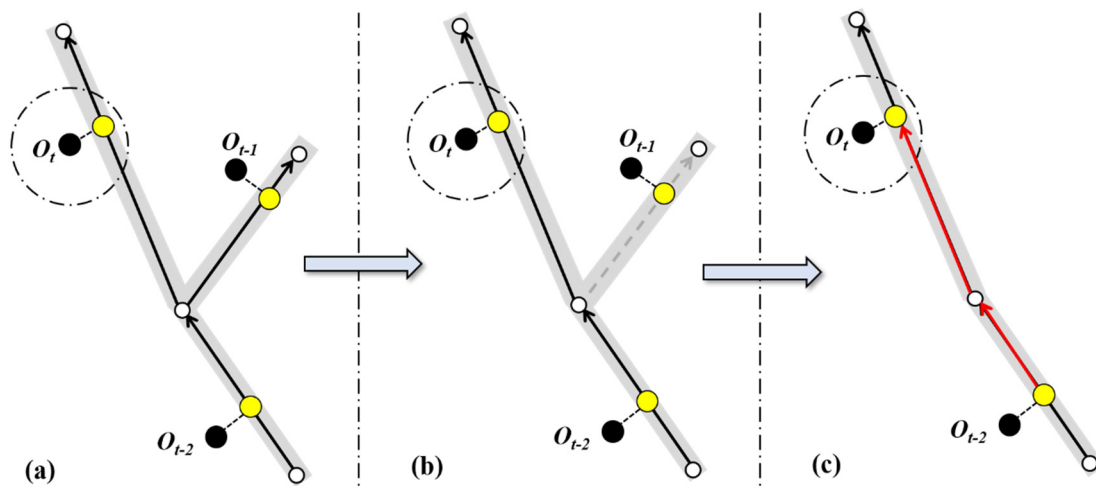


Figure 3. Example of delay matching strategy

3. Experiment

3.1. Data

The trajectory data used in our experiment comes from the open trajectory data uploaded by users on

OpenStreetMap (<https://www.openstreetmap.org>). The sampling interval is 1s. Route 1 contains 761 trajectory points, and route 2 contains 516 trajectory points. Each record contains mainly timestamp, longitude and latitude, no speed, direction, etc. Road network data are also obtained from OpenStreetMap, and its range can cover the entire trajectory. Taking Route 1 as an example (Figure 4), it can be preliminarily found that there are obvious deviations in some segments of the road network through discrimination. There are typical roads in the road network, such as single road, double road, crossroad, and roundabout, which will help us to better test the algorithm.

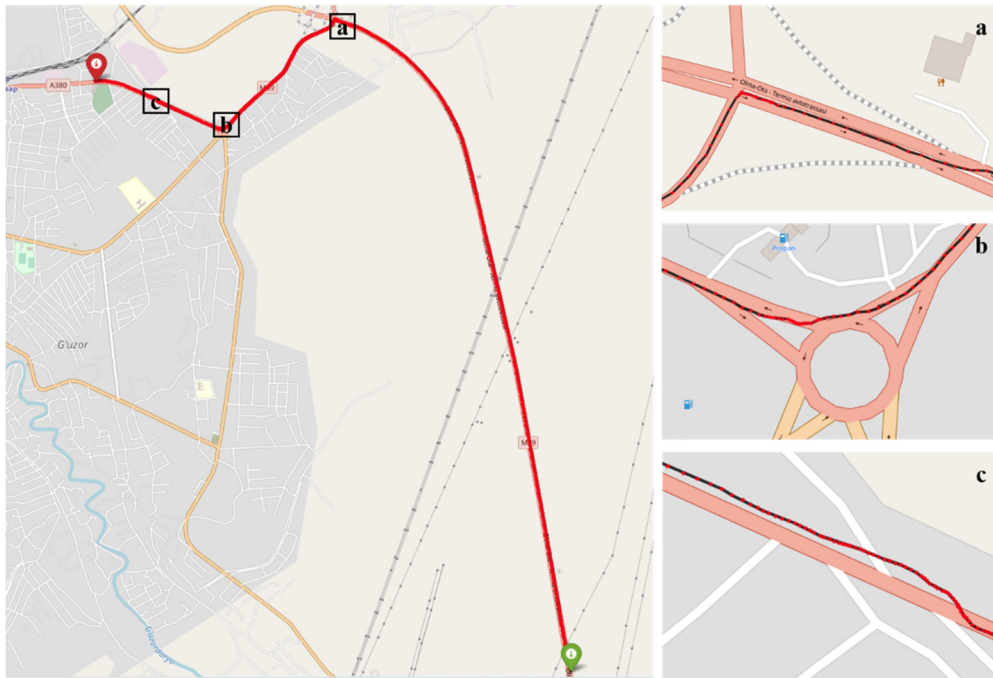


Figure 4. Trajectory data of route 1

3.2. Data preprocessing

To facilitate subsequent calculation and analysis, we convert trajectory data and road network data to the same projection coordinate system (e.g. UTM) before data input. In addition, the road network directed graph is constructed to facilitate the search of the shortest path and the calculation of the path length in the matching process.

3.3. Evaluation metrics

The effect of the algorithm is evaluated by matching accuracy A_L and output delay T . The calculation formula is as follows:

$$A_L = (1 - \frac{|L_m - L_t|}{L_t}) \times 100\% \quad (3)$$

$$T = \frac{\sum_{i=1}^N T_i}{N} \quad (4)$$

Where L_m is the length of the output path after matching, L_t is the corresponding real road path length, T_i is the time required to match a single trajectory point, and N is the number of trajectory points processed.

3.4. Results

Figure 5 shows some matching details of route 1. It shows that the original trajectory appears across the single road (Figure 5(a)), which is not allowed to occur under the constraints of traffic rules. The

algorithm corrects this part of the trajectory well, so that the trajectory extends normally on the single road and turns at the intersection. In addition, the trajectories at the roundabout and double road are also correctly matched to the corresponding roads (Figure 5(b), Figure 5(c)).

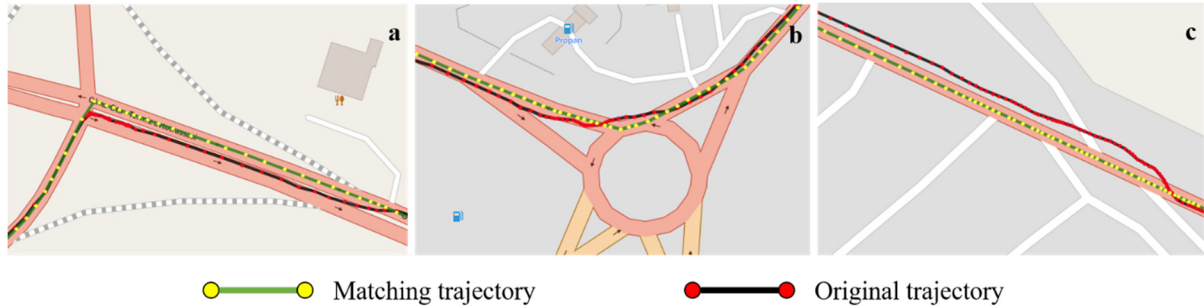


Figure 5. Partial matching results of route 1

To test the matching effect of the algorithm on trajectories with different sampling rates, the original trajectory data are thinned to obtain the data with sampling rates of 1s, 10s, 30s, and 60s, respectively. Table 1 shows the A_L and T of route 1 and route 2 at different sampling rates. It shows that the matching accuracy of the algorithm under different sampling rates is high, and it can meet the low delay output requirements of online matching.

Table 1. A_L and T under different sampling rates

Sampling rate	A_L		T (ms)	
	Route1	Route2	Route1	Route2
1s	99.85%	99.86%	108	114
10s	99.75%	99.21%	110	112
30s	99.34%	98.60%	107	111
60s	96.03%	93.15%	107	108
Average	98.74%	97.71%	108	111

4. Conclusions and future work

In this paper, we propose a self-adjusting online map matching method, called AOMM. The effect of the algorithm is evaluated by using the open trajectory data, and the map matching experiments under different sampling rates are designed by thinning the trajectory data. The results show that the algorithm has high matching accuracy and low output delay, which can meet the needs of online matching. In addition, the input data of the algorithm are simple, and the adjustment strategy provided can better ensure the stability and matching accuracy of the algorithm.

The accuracy of online map matching usually depends on the matching result of the historical trajectory, so it is easy to produce a cumulative error when the historical trajectory is a suboptimal matching scheme. The output delay of online matching is not only related to the algorithm, but also affected by the computer performance and operating environment. In addition, richer data input will be conducive to the improvement of map matching accuracy, but it may also increase the complexity of the algorithm. In future work, we can further explore solutions to these problems.

References

- [1] Pfoser, D., Jensen, C.S. (1999) Capturing the uncertainty of moving-object representations. In: International Symposium on Advances in Spatial Databases. Berlin. pp. 111-131.
- [2] Joshi, R.R. (2001) A new approach to map matching for in-vehicle navigation systems: the rotational variation metric. In: IEEE Intelligent Transportation Systems Conference. Oakland. pp. 33-26.
- [3] Wei, L.Y., Zheng, Y., Peng, W.C. (2012) Constructing popular routes from uncertain trajectories. In: 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Beijing. pp. 195-203.

- [4] Yuan, J., Zheng, Y., Xie, X., et al. (2013) T-Drive: Enhancing driving directions with taxi drivers' intelligence. *IEEE Transactions on Knowledge and Data Engineering*, 25: 220-232.
- [5] Mori, U., Mendiburu, A., Lvarez, M., et al. (2015) A review of travel time estimation and forecasting for advanced traveller information systems. *Transportmetrica A: Transport Science*, 11: 119-157.
- [6] Krumm, J. (2016) A markov model for driver turn prediction. *Sae World Congress*, 22: 1-25.
- [7] Bernstein, D., Kornhauser, A. (1996) An introduction to map matching for personal navigation assistants. *Geometric Distributions*, 122: 1082-1083.
- [8] Cew, A., Db, B., Alk, A. (2000) Some map matching algorithms for personal navigation assistants - ScienceDirect. *Transportation Research Part C: Emerging Technologies*, 8: 91-108.
- [9] Taylor, G., Blewitt, G., Steup, D., et al. (2001) Road reduction filtering for GPS-GIS navigation. *Transactions in GIS*, 5: 193-207.
- [10] Alt, H., Efrat, A., Rote, G., et al (2003) Matching planar maps. *Journal of Algorithms*, 49: 262-283.
- [11] Quddus, M., Ochieng, W.Y., Lin, Z., et al (2003) A general map matching algorithm for transport telematics applications. *GPS Solutions*, 7: 157-167.
- [12] Jabbour, M., Bonnifait, P., Cherfaoui, V. (2008) Map-matching integrity using multi-sensor fusion and multi-hypothesis road tracking. *Journal of Molecular Biology*, 12: 189-201.
- [13] Noland, R.B., Quddus, M., Ochieng, W.Y. (2003) Map-matching in complex urban road networks. *Revista Brasileira de Cartografia*, 55: 1-14.
- [14] Su, H., Chen, J., Xu, J. (2008) A adaptive map matching algorithm based on Fuzzy-Neural-Network for vehicle navigation system. In: *Proceedings of the 7th world congress on intelligent control and automation*. Chongqing. pp. 25-27.
- [15] Li, Z., Wu, C. (2005) A new approach to map-matching and parameter correcting for vehicle navigation system in the area of shadow of GPS signal. In: *IEEE Conference on Intelligent Transportation Systems*. Vinnea. pp. 425-430.
- [16] Smaili, C., Najjar, M., Charpillat, F. (2008) A road matching method for precise vehicle localization using hybrid Bayesian network. *Journal of Intelligent Transportation Systems*, 12: 176-188.
- [17] Lou, Y., Zhang, C., Zheng, Y., et al. (2009) Map-Matching for low-sampling-rate GPS trajectories. In: *ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems*. Washington. pp. 352-361.
- [18] Liu, X., Liu, K., Li, M., et al. (2017) A ST-CRF Map-Matching Method for Low-Frequency Floating Car Data. *IEEE Transactions on Intelligent Transportation Systems*, 18: 1241-1254.
- [19] Gong, Y.J., Chen, E., Zhang, X., et al. (2018) AntMapper: An Ant Colony-Based Map Matching Approach for Trajectory-Based Applications. *IEEE Transactions on Intelligent Transportation Systems*, 19: 390-401.
- [20] Wenk, C., Salas, R., Pfoser, D. (2006) Addressing the Need for Map-Matching Speed: Localizing Global Curve-Matching Algorithms. In: *18th International Conference on Scientific and Statistical Database Management*. Vienna. pp. 879-888.
- [21] Liang, B., Li, S., Li, S., et al. (2016) Online Learning for Accurate Real-Time Map Matching. In: *20th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. New York. pp. 67-78.
- [22] Rohani, M., Gingras, D., Gruyer, D. (2015) A novel approach for improved vehicular positioning using cooperative map matching and dynamic base station DGPS concept. *IEEE Transactions on Intelligent Transportation Systems*, 17: 1-10.
- [23] Yuan, J., Zheng, Y., Zhang, C., et al. (2010) An Interactive-Voting Based Map Matching Algorithm. In: *11th International Conference on Mobile Data Management*. Kansas. pp. 43-52.
- [24] Brakatsoulas, S., Pfoser, D., Tryfona, N. (2005) On map-matching vehicle tracking data. In: *31st International Conference on Very Large Data Bases*. Trondheim. pp. 853-864

- [25] Goh, C.Y., Dauwels, J., Mitrovic, N., et al. (2012) Online map-matching based on Hidden Markov model for real-time traffic sensing applications. In: 15th International IEEE Conference on Intelligent Transportation Systems. Anchorage. pp: 776-781.
- [26] Newson, P., Krumm, J. (2009) Hidden Markov map matching through noise and sparseness. In: 17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems. Washington. pp: 336-343.
- [27] Jagadeesh, G.R., Srikanthan, T. (2017) Online Map-Matching of Noisy and Sparse Location Data with Hidden Markov and Route Choice Models. IEEE Transactions on Intelligent Transportation Systems, 18: 2423-2434.
- [28] Viterbi, A.J. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, 13: 260-269.
- [29] Luo, A., Chen, S.H., X, B. (2017) Enhanced Map-Matching Algorithm with a Hidden Markov Model for Mobile Phone Positioning. ISPRS International Journal of Geo-Information, 6: 327-327.
- [30] Hart, P.E., Nilsson, N.J., Raphael B. (1968) A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions of Systems Science and Cybernetics, 4: 100-107.