BMI-Matching: Map-Matching with Bearing Meta-information

DaWei Wang and JingJing Gu(✉)

College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, Nanjing, China
wangdawei673@163.com, gujingjing@nuaa.edu.cn

Abstract. Map-matching is a fundamental pre-processing step for many applications which aligns a trajectory represented by a sequence of sampling points with the city road network on a digital map. With the help of GPS-embedded devices, a lot of GPS trajectories can be collected. However, the raw positions captured by GPS devices usually can not reflect the real positions because of physical constraints such as GPS signals blocked by buildings. And low-sampling-rate data is another challenge for map-matching. Although many approaches have been proposed to solve these problems, unfortunately, most of them only consider the position of the object or the topology structures of the road network. So it becomes significant to accurately match GPS trajectories to road network. We propose a method called BMI-matching (map-matching with bearing meta-information) which not only considers the two factors above but also focuses on the moving object bearing. Based on bearing, we can calculate the direction similarity between moving object and road segments to determine selecting which road segment is appropriate. We conduct experiments on real dataset and compare our method with two state-of-the-art algorithms. The results show that our approach gets better performance on matching accuracy.

Keywords: Map-matching · HMM · R-tree

1 Introduction

In recent years, with the advance of diverse location-acquisition technologies, mining object trajectories have attracted lots of researcher's attention. Map-matching deals with the problem of matching a series of GPS points to the city roads on a digital map. It is fairly useful for many applications, such as understanding urban mobility [8], discovering critical nodes in road network [19], popular routes finding [17], exploring the Urban Region-of-Interest [16] and travel plan recommendation [9]. Nevertheless, the raw position data collected by GPS devices may not report the real position of moving objects. Noisy data and low sampling rate data [14] are usually the big challenges for map-matching problem. The positions measured by GPS devices are not accurate. Especially when vehicle enters a tunnel or an urban canyon, the GPS is experiencing particularly high

X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 712–727, 2019.
https://doi.org/10.1007/978-3-030-32388-2_58

noise. GPS error can be described as a two dimensional Gaussian distribution. In Fig. 1, if we regard the sampling position as the origin, the distribution of real position is an ellipse or a circle, which depends on the variance and expectation of each dimensionality and correlation coefficient of two dimensionalities. The height represents the possibility of the real position. High noise means the real position is far away from the sampling point and makes map-matching more difficult. Low sampling rate may lead to the uncertainty of an object's moving track. Let us consider two sampling points p_1, p_2. If the sampling time interval is t and the object moving speed is v, the moving distance between these two points is $l = v * t$. In Fig. 2, p_1, p_2 are two foci of the ellipse, and $l_1 = l_2 = l$ are two paths where object may travel. Generally, if the sampling rate is low (large t), the number of possible paths will be large, which means huge uncertainty of track between these two points.

And we perform an analysis on the taxi trajectory dataset collected in Shanghai, China. The taxis often report their positions to the dispatching center with low sampling rates for saving energy and communication cost. The sampling rates usually vary from a few seconds to minutes. Owing to kinds of factors (sensor failure, transmission error, etc.), the location reported by devices may be noisy. As shown in Fig. 3, the sampling time interval in our data exceeds 2 min. In Fig. 4, the green lines represent road network, and the trajectory is represented by yellow points. Obviously, the trajectory misses many sampling points. It will be challenging to match these data on road network.

A number of approaches have been proposed for map-matching problem. Some conventional map-matching methods employ local or incremental algorithms [4] to map current or neighboring positions. But it may fail for low sampling rate data. A few methods match the trajectory exploiting global relationship to deal with the low sampling rate data [7,17]. But they can not perform well on complex environments such as thick road network. Although other methods for low-sampling-rate data [10,11] and noisy data [11] perform well, they need particular conditions such as complete information of road network which may not be available in practice.

Overall, in this paper, we make three contributions summarized as follows.

- We propose a method called BMI-matching which considers the bearing of the moving object besides the location of object and the topology structures of the road network. Based on moving object bearing, we can compare the direction similarity between moving object and road segments.
- In the complex road network, it is hard to query the candidate segments in a short time. We exploit spatial access method R-tree to accelerate the search process. Calculating shortest path is computationally high. We compute shortest paths covering hotspots in advance and save as an index table.
- We conduct experiments on real dataset and compare our algorithm with two state-of-the-art algorithms of ST-Matching [10] and HMM-Matching [11] in matching accuracy. The results show that the our algorithm gets better performance on accuracy.

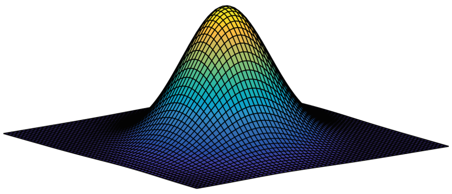


Fig. 1. Illustration of noisy data

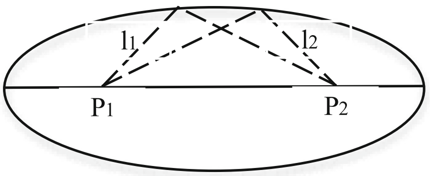
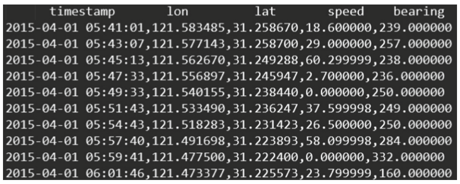


Fig. 2. Illustration of low sampling rate data



```
          timestamp         lon          lat         speed     bearing
2015-04-01 05:41:01,121.583485,31.258670,18.600000,239.000000
2015-04-01 05:43:07,121.577143,31.258700,29.000000,257.000000
2015-04-01 05:45:13,121.562670,31.249288,60.299999,238.000000
2015-04-01 05:47:33,121.556897,31.245947,2.700000,236.000000
2015-04-01 05:49:33,121.540155,31.238440,0.000000,250.000000
2015-04-01 05:51:43,121.533490,31.236247,37.599998,249.000000
2015-04-01 05:54:43,121.518283,31.231423,26.500000,250.000000
2015-04-01 05:57:40,121.491698,31.223893,58.099998,284.000000
2015-04-01 05:59:41,121.477500,31.222400,0.000000,332.000000
2015-04-01 06:01:46,121.473377,31.225573,23.799999,160.000000
```

Fig. 3. The form of taxi data

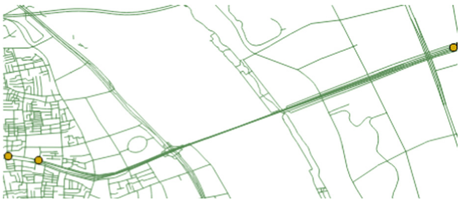


Fig. 4. A trajectory visualized on digital map (Color figure online)

2 Related Work

Researchers pay more attention on mining moving objects trajectories to understand resident mobility, discover the functionality of different regions in a city [16] and construct the smart city. The fundamental step of all these applications is map-matching. A number of map-matching algorithms have been developed using different techniques. There are two approaches to classify map-matching methods, based on additional information used, or the range of sampling points considered in a trajectory.

According to the range of sampling points, map-matching algorithms can be classified into two groups: local/incremental and global methods. Some conventional map-matching methods employ local or incremental algorithms [4] to map current or neighboring positions. These approaches are fast in computation and work well when sampling frequency is very high. However, their performance is susceptible to the decrease of sampling frequency. Global algorithms aim to match an entire trajectory with a road network. Usually global methods are more accurate than local methods, but it may take long time and is often applied to offline tasks.

From the perspective of additional information used, map-matching algorithms can be classified into four categories: geometric, topological, probabilistic and advanced techniques. For geometric algorithms, they exploit the geometric information of the road network by only considering the shape of the links [2,4], such as matching a GPS point to the nearest road segment. A topological method for map-matching pays attention to the connectivity of the roads. The method proposed in [20] aims to find a minimum weight path based on edit distance. Method proposed in [1] utilizes the Fréchet distance to measure the fit between a GPS sequence and candidate road segment sequence. Probabilistic methods consider various error sources associated with the navigation sensor and the road network data quality [13]. Other advanced algorithms use Kalman filter [12], fuzzy logic [15], or the application of Hidden Markov Model [11]. Some variants of HMM based algorithm have been proposed in [10,21]. The ST-Matching algorithm proposed in [10] combines spatial analysis and temporal analysis which is based on the speed constraint of the road. However, the speed constraint of the road is the maximum speed, and moving objects have different travel speeds at different time of the day. Our algorithm is inspired by the HMM algorithm and considers the bearing of the moving objects.

3 Preliminary Knowledge

In this section, we will give some basic definitions.

Definition 1. *A* ***sampling point****(p) is a tuple denoted as <lon, lat, speed, bearing, timestamp>, which stands for the longitude, latitude, speed, moving direction of object and generation time of p.*

For representing the bearing, a basis and a positive direction are needed. We define the north as the basis and the clockwise as the positive direction. The bearing is the angle between the basis and the moving direction. As shown in Fig. 5, Y-axis denotes the basis and the moving direction of sampling point p_t is $\boldsymbol{d}$, so the bearing can be represented as the angle θ ($\theta \in [0, 2\pi]$).

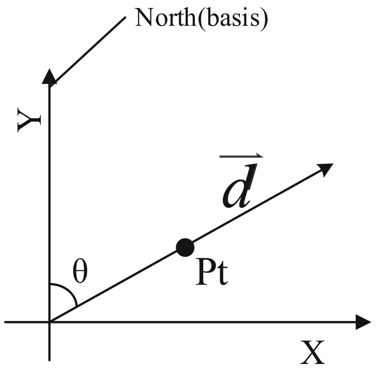


Fig. 5. The bearing of sampling point p at time t

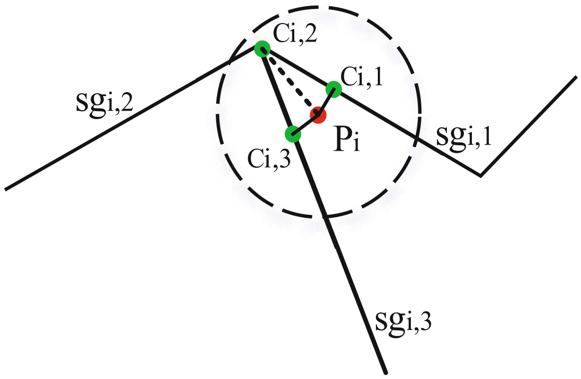


Fig. 6. Candidate segments and segment projection points (Color figure online)

Definition 2. *A **GPS trajectory**(tr) is a series of sampling points with the time interval between any consecutive sampling points not exceeding a certain threshold $\varDelta T$. $tr = \{p_1, p_2, ..., p_n\}$,where p_1 is the start point and p_n is the end point.*

Definition 3. *A **segment**(sg) is a directed edge with two terminal points($sg.sp, sg.ep$) on road network, where $sg.sp$ is the start point and $sg.ep$ is the end point. Each segment also has two attributes of ID($sg.ID$) and length($sg.l$) .*

Definition 4. *A **road network** is a directed graph $G(V, E)$, where V is a set of nodes representing the intersections and terminal points of the road segments, and E is a set of directed edges representing road segments.*

Definition 5. *A **route**(r) is a set of segments. $r = \{sg_1, sg_2, ..., sg_n\}$, where $r.sp = sg_1.sp$ and $r.ep = sg_n.ep$*

Definition 6. *A **segment projection point**($c_{i,j}$) of a sampling point(p_i) on the segment($sg_{i,j}$) is a point such that $c_{i,j} = arg\ min_{\forall k_m \in sg_{i,j}}\ dist(p_i, k_m)$, where k_m is any point on $sg_{i,j}$ and $dist(p_i, k_m)$ is the distance between the sampling point and the point in segment.*

As shown in Fig. 6, red point and green points represent sampling point and segment projection points respectively, and segments $sg_{i,1}, sg_{i,2}, sg_{i,3}$ are candidate segments within radius R.

4 Framework

With the preliminary knowledge above, our improved algorithm is presented in this section. The application logic of our method is demonstrated in Fig. 7.

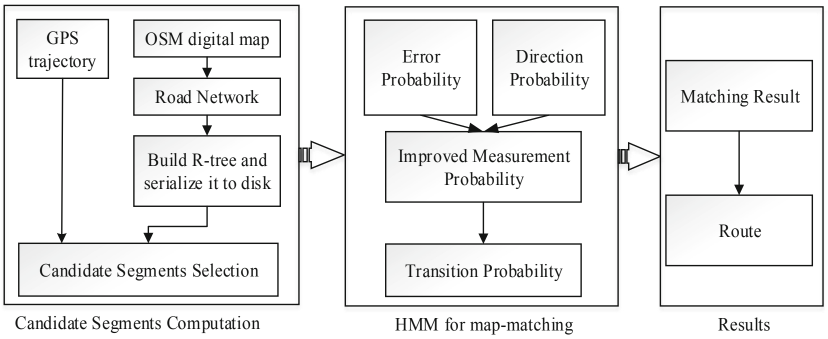
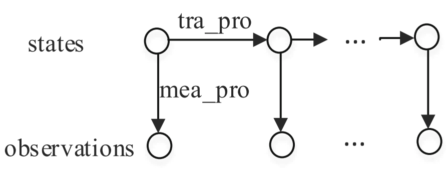


Fig. 7. System overview

HMM for map-matching: a hidden Markov model can be considered a generalization of a mixture model where the hidden variables, which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other. As shown in Fig. 8, the observations are time sequence data and each observation may involve many possible states. there is a transition probability(tra_pro) between states and a measurement probability(mea_pro) between observation and state. Hidden Markov models are especially known for their application in temporal pattern recognition such as speech recognition, machine translation, gene prediction and so on.

As illustrated in Fig. 9, there is a trajectory marked with a sequence of sampling points $p_1, p_2, ..., p_n$ called observations and each point has a list of candidate road segments $sg_1, sg_2, ..., sg_m$ called states. The goal of HMM algorithm in map-matching problem is to find an optimal path in many feasible paths by picking one road segment for each sampling point.



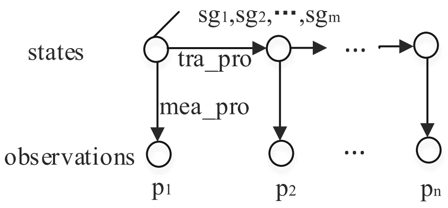


Fig. 8. HMM model

Fig. 9. HMM for map-matching

Candidate Segments Selection: first, this component accepts a GPS trajectory as an input. Then, it retrieves all candidate segments within radius R and adjusts the number of candidate segments according to N for each sampling point. Next, segment projection point is computed on each segment. It finally outputs the list of candidate segments and segment projection points.

Improved measurement probability: it considers the bearing of the moving object and the distance between the sampling point and the coordinate segment projection point.

Transition probability: we assume that the driver would follow the shortest path to obtain maximum interest. Some transitions will be more likely and vice versa. For instance, different segments with the same way ID or much closed to each other have high transition probability.

Matching result: this component evaluates all possible routes using probability information and gives the optimal route for the GPS trajectory.

5 Algorithm Details

In this section, we show our BMI-matching algorithm in details.

Candidate Segments Selection: given a GPS trajectory $tr = \{p_1, p_2, ..., p_n\}$, we retrieve all candidate segments of each sampling point p_i within radius R. Then we compute the segment projection point on each segment. Sometimes the city road network may be large and complicated, so it is hard to retrieve all candidate segments of each sampling point in a short time. To speed up the algorithm,we can use spatial access methods R-tree [5]. Because there are enough candidate segments within radius R, we set another parameter, number of candidate segments N, to reduce the execution time.

As shown in Fig. 6, the sampling point p_i has three candidate segments $sg_{i,1}, sg_{i,2}$ and $sg_{i,3}$. Projection point is computed simultaneously. The geometry projection point of p_i onto segment $sg_{i,2}$ is beyond endpoints, and we choose the nearest point $c_{i,2}$ as the projection point. Algorithm 1 shows the detailed procedure.

Improved measurement probability: in improved measurement probability, we make full use of geometric information of the road network and the bearing of the moving object. For map-matching, given a location of the sampling point p_t, there is an error (GPS error) probability $N(p_t|sg_i)$ and a direction probability $D(p_t|sg_i)$ for each candidate segment sg_i. We can model GPS error as normal distribution $N(\mu, \delta^2)$ based on previous work [3]. Formally, we define error probability $N(p_t|sg_i)$ of the sampling point p_t for the candidate segment sg_i as

$$N(p_t|sg_i) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{(x_t^i - \mu)^2}{2\delta^2}} \tag{1}$$

where x_t^i is the distance between p_t and segment projection point $c_{t,i}$ on segment $sg_{t,i}$. We can easily determine the road segment when comparing the direction of moving object to the direction of road segment. We define the direction probability $D(p_t|sg_i)$ of the sampling point p_t for the segment sg_i as

$$D(p_t|sg_i) = log(1 + exp(-a))$$
$$a = min(\ |diff(\Theta_{p_t}, \Theta_{sg_i})|,\ 2\pi - |diff(\Theta_{p_t}, \Theta_{sg_i})|\ )$$

where Θ_{p_t} and Θ_{sg_i} are the bearing of moving object at time t and road segment sg_i respectively, and $diff(\Theta_{p_t}, \Theta_{sg_i})$ is the difference of bearing between them. With the error probability $N(p_t|sg_i)$ and direction probability $D(p_t|sg_i)$, we define the improved measurement probability of the sampling point p_t for the segment sg_i as

$$I(p_t|sg_i) = N * D \tag{2}$$

Algorithm 1. Candidate segments Selection

Pre-processing: build the R-tree of the city road network and serialize it to disk.

Input: road network $G < V, E >$, GPS trajectory tr, radius R, number of candidate segments N

Output: list of candidate segments SG and segment projection points C

```
1:  for each point $p_i$ in $tr$ do
2:      retrieve the all candidate segments as a list $SG_i$ in $G$ within $R$
3:      if $len(SG_i) > N$ then
4:          select $N$ segments $sg_{i,n}$ from $SG_i$.
5:          compute projection point in each segments as a list $c_{i,n}$.
6:          $SG.append(sg_{i,n})$
7:          $C.append(c_{i,n})$
8:      if $0 < len(SGi) < N$ then
9:          compute projection point in each segments as a list $C_i$.
10:         $SG.append(SG_i)$
11:         $C.append(C_i)$
12: return SG,C
```

Initial state probability: for map-matching, initial state probablity π gives the probability of the vehicle's first road segment over all segments. With the definition of improved measurement probability, we describe the initial state probability as the first improved measurement probability:

$$\pi_i = I(z_1|sg_i) = N(z_1|sg_i) * D(z_1|sg_i) \tag{3}$$

In practice, it is less possible to match the sampling point into the road segment which is far from it. So, we set to zero any error probability from a road segment that is more than radius R away from the sampling point.

Transition probability: given two sampling point p_t and p_{t+1}, transition probability is the probability of a vehicle driving between two candidate road segments at these two times. We assume that the driver would follow the shortest path to obtain maximum interest. The route distance will be computed by method of shortest path. Thus, the transitions whose great circle distance is about the route distance between two sampling points may be more possible. As shown in Fig. 10, there are four segments sg_1, sg_2, sg_3 and sg_4 and two sampling points p_t and p_{t+1}. $c_{t,1}, c_{t,2}$ and $c_{t+1,1}$ are projection points of the sampling point p_t and p_{t+1} respectively. Red dashed line represents the great circle distance between p_t and p_{t+1} denoted as $greatCD$. Green line and orange line represent the route distance denoted as $route1$ and $route2$ respectively. $greatCD$ is much closer to $route2$ rather than $route1$. In other words, the true path is $route2$. We define the transition probability as

$$p(sg_{t+1}|sg_t) = e^{-d_t}$$
$$d_t = |greatCircleDis(p_t, p_{t+1}) - routeDis(c_{t,i}, c_{t+1,j})|$$

where sg_t and sg_{t+1} are the candidate segments of the sampling point p_t and p_{t+1} at these two times respectively. The route distance can be calculated by shortest path methods such as Dijkstra, A* and so on, but all of them are computationally high. In practice, we compute some shortest paths covering hotspots in advance and save it to file as an index table. We can check the shortest paths file when needed.

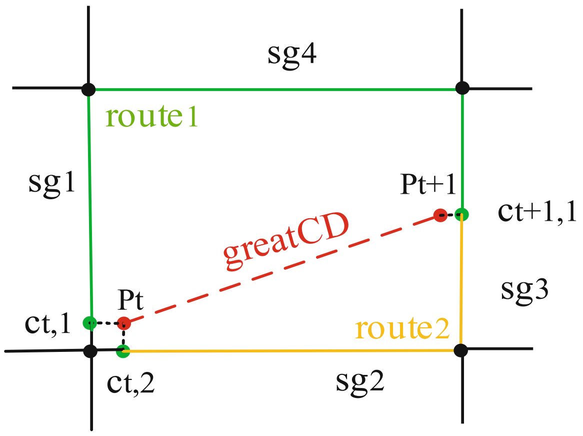


Fig. 10. Transition probability (Color figure online)

Matching result: with improved measurement probability, initial state probability and transition probability, we can compute the optimal path using the Viterbi algorithm. The Viterbi algorithm is a dynamic programming algorithm to maximize the product of the improved measurement probability and the transition probability for finding the most likely sequence of hidden states.

6 Experiment

In this section, we present the experimental evaluation and verify that our algorithm can achieve high accuracy. First we show the experimental setting, including the dataset and some parameters. Next we introduce two evaluation criteria. Then we compare our algorithm with two algorithms of the HMM-Matching proposed by Newson et al. [11] and ST-Matching proposed by Lou et al. [10].

6.1 Experimental Setting

Dataset

Road Network: we use Shanghai road network which can be obtained by OpenStreetMap (OSM). OSM is a collaborative project to create a free editable map of the world [6,18]. As depicted in Fig. 11, the road network(highway) of ShangHai includes 694,572 vertices and 887,153 road segments.

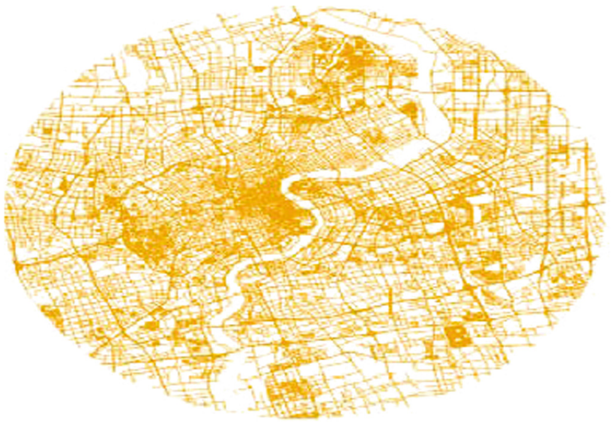


Fig. 11. A part of road network in Shanghai

Taxi Trajectory Data: we collect 13,636 taxis' trajectories for one month in Shanghai. We also select eight trajectories which cover not only downtown area but also suburban area as groundtruth.

Parameters

In our algorithm, there are some parameters that need to be set, including the search radius R in candidate segments selection, expectation μ and standard deviation δ in error probability. And we will turn these parameters according to the evaluation criteria defined next. We also compare our algorithm with HMM-Matching and ST-Matching as two baselines. For these two baseline algorithms, we use the empirical settings $\mu_b = 0, \delta_b = 25, \beta_b = 0.16$ as suggested in [10,11].

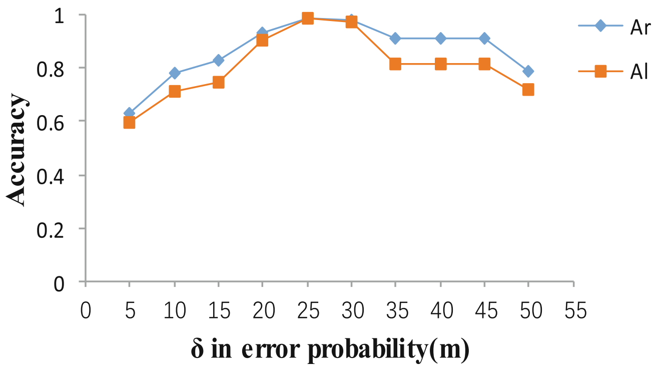


Fig. 12. The effect of δ in error probability

Evaluation Criteria
Our algorithm is evaluated both in terms of running time and matching accuracy. The actual program execution time is the measurement for running time. The matching accuracy is measured by two criteria A_r, A_l, computed by following equations:

$$A_r = \frac{number\ of\ correctly\ matched\ road\ segments}{number\ of\ all\ road\ segments\ of\ a\ test\ trajectory}$$

$$A_l = \frac{\Sigma\ length\ of\ correctly\ matched\ road\ segments}{length\ of\ a\ test\ trajectory}$$

6.2 Experimental Results

Error Probability Parameters μ, δ
GPS error follows the zero-mean Gaussian, so we set $\mu = 0$. We fix the other parameters by assigning the search radius R as 200 meters and number of candidate segments N as 50 and tuning δ.

As shown in Fig. 12, A_r and A_l increase with δ increasing when δ is smaller than 25 m. Ar and Al decrease with δ increasing when δ is large than 25 m. A small value of δ means we have more confidence in GPS device and the candidate segments nearby the sampling points will be selected as the matched segments in a high probability. However, when δ is large the effect of error probability will be diminished. The results of the experiment demonstrate that zero-mean normal distribution with the standard deviation of 25 m is suitable for our algorithm.

Running Time Evaluation
To test the actual program execution time of our algorithm, we select 6 trajectories with different lengths (number of sampling points) which cover different regions.

As depicted in Fig. 13, the execution time will increase with the number of sampling points increasing generally, but there are not much differences between the fifth and sixth trajectory in terms of running time. We find the road network around the sixth trajectory is not complicated when visualizing it on the digital map, which means the number of candidate segments for some sampling points are small. So the running time depends on not only the length of the trajectory but also the number of candidate segments.

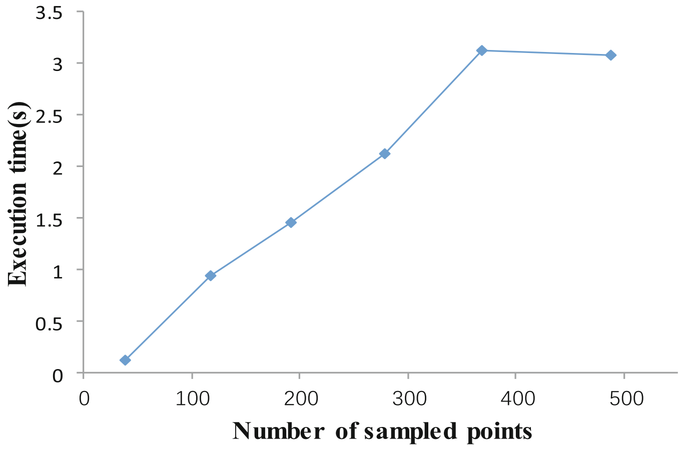


Fig. 13. The execution time with different number of sampling points

Number of Candidate Segments *N* and Search Radius *R*
We set search radius $R = 50$, because the δ in error probability is 25 m. Search radius R and number of candidate segments N determine the candidate segments and projection points. When N is small the accuracy will decrease, and a large N may lead to high program running time. So an appropriate value of N is required.

As shown in Fig. 14, A_r and A_l increase with N increasing when N is smaller than 5. A_r and A_l become steady when N exceeds 5. And we set $N = 5$ to get better performance on accuracy without much time cost.

Matching Result Comparison
We select eight groundtruth trajectories and compare our algorithm with HMM-Matching and ST-Matching based on two criteria A_r and A_l. In Figs. 15 and 16, we can find our algorithm gets higher accuracy compared to HMM-Matching and ST-Matching both on A_r and A_l.

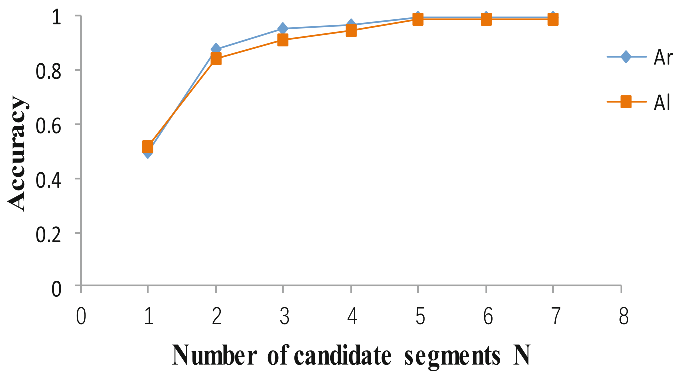


Fig. 14. The effect of number of candidate segments N

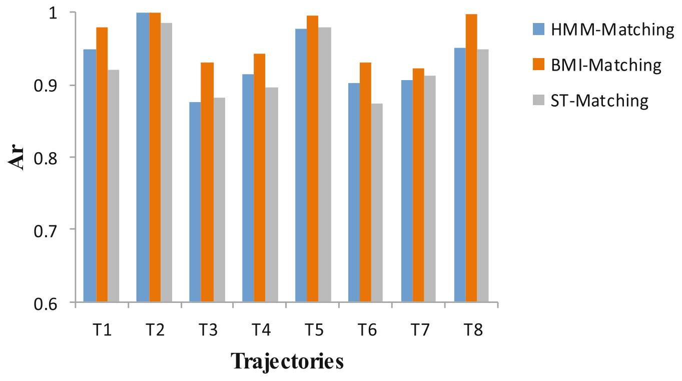


Fig. 15. The comparison on A_r

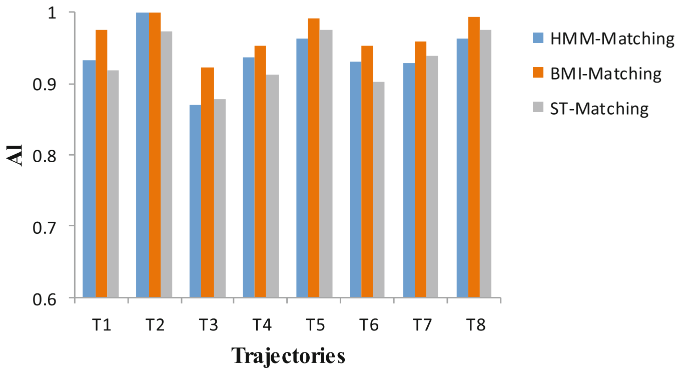


Fig. 16. The comparison on A_l

Our algorithm works well in some cases compared with the HMM-Matching algorithm and we don't need to tune parameters in transition probability. As

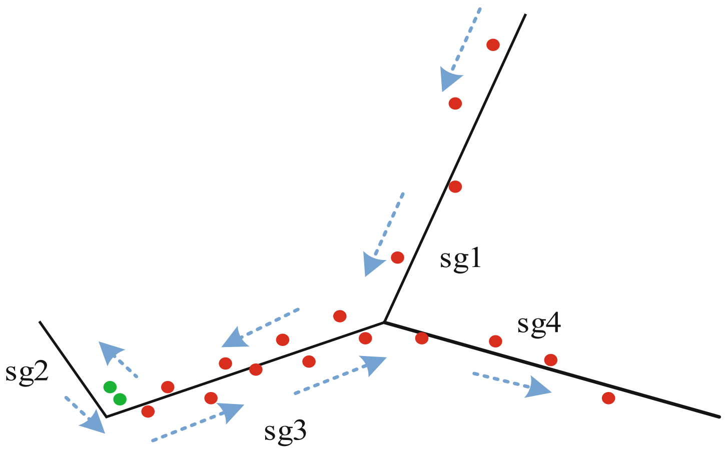


Fig. 17. Case1 (Color figure online)

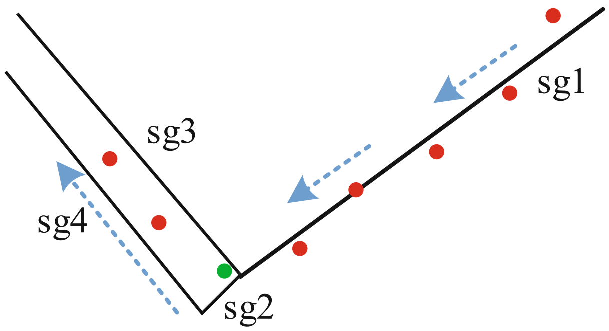


Fig. 18. Case2 (Color figure online)

depicted in Fig. 17, there are four road segments sg_1, sg_2, sg_3, sg_4. Blue arrows represent the direction of moving object. The true path is $sg_1 \longrightarrow sg_2 \longrightarrow sg_3 \longrightarrow sg_4$. The bearing of green points and raod segment sg_2 are similar, so we believe matching green points into $sg2$ is appropriate. But the HMM-Matching algorithm will match them into sg_3 instead of sg_2, which is not correct. In other words, the HMM-Matching algorithm will make errors in a short turn sometimes. In Fig. 18, the distance between last two red points and sg_3 or sg_4 is same, hence matching the green point into segment correctly becomes crucial. Because the green point is near to sg_3, HMM-Matching algorithm matchs it into sg_3 without considering the bearing.

7 Conclusion

In this paper, we introduce a BMI-Matching algorithm. We make full use of the position and bearing of moving objects and the topological structures of road network. We conduct experiments with a real dataset and perform the comparisons between our method and two algorithms of ST-Matching and HMM-Matching. The results show our method performs better than those algorithms in terms of A_r and A_l.

References

1. Brakatsoulas, S., Pfoser, D., Salas, R., Wenk, C.: On map-matching vehicle tracking data. In: International Conference on Very Large Data Bases (2005)
2. Chen, D., Driemel, A., Guibas, L.J., Nguyen, A., Wenk, C.: Approximate map matching with respect to the Fréchet distance. In: 2011 Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments (ALENEX), pp. 75–83. SIAM (2011)
3. Diggelen, F.V.: GNSS accuracy: lies, damn lies, and statistics. Gps World, January 2007
4. Greenfeld, J.S.: Matching GPS observations to locations on a digital map. In: 81th Annual Meeting of the Transportation Research Board, vol. 1, pp. 164–173 (2002)
5. Guttman, A.: R-trees: a dynamic index structure for spatial searching, vol. 14. ACM (1984)
6. Haklay, M., Weber, P.: OpenStreetMap: user-generated street maps. IEEE Perv. Comput. **7**(4), 12–18 (2008)
7. Kai, Z., Yu, Z., Xing, X., Zhou, X.: Reducing uncertainty of low-sampling-rate trajectories. In: IEEE International Conference on Data Engineering (2012)
8. Kumar, D., Wu, H., Rajasegarar, S., Leckie, C., Krishnaswamy, S., Palaniswami, M.: Fast and scalable big data trajectory clustering for understanding urban mobility. IEEE Trans. Intell. Transp. Syst. **99**, 1–14 (2018)
9. Liu, H., Li, T., Hu, R., Fu, Y., Gu, J., Xiong, H.: Joint representation learning for multi-modal transportation recommendation. In: AAAI (2019, to appear)
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: ACM Sigspatial International Conference on Advances in Geographic Information Systems (2009)
12. Obradovic, D., Lenz, H., Schupfner, M.: Fusion of map and sensor data in a modern car navigation system. J. VLSI Sig. Process. Syst. Sig. Image Video Technol. **45**(1–2), 111–122 (2006)
13. Ochieng, W.Y., Quddus, M.A., Noland, R.B.: Map-matching in complex urban road networks (2003)
14. Pfoser, D., Jensen, C.S.: Capturing the uncertainty of moving-object representations. In: Güting, R.H., Papadias, D., Lochovsky, F. (eds.) SSD 1999. LNCS, vol. 1651, pp. 111–131. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48482-5_9
15. Quddus, M.A., Noland, R.B., Ochieng, W.Y.: A high accuracy fuzzy logic based map matching algorithm for road transport. J. Intell. Transp. Syst. **10**(3), 103–115 (2006)

16. Sun, Y., Zhu, H., Zhuang, F., Gu, J., He, Q.: Exploring the urban region-of-interest through the analysis of online map search queries. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2269–2278. ACM (2018)
17. Wei, L.Y., Zheng, Y., Peng, W.C.: Constructing popular routes from uncertain trajectories. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 195–203. ACM (2012)
18. Wikipedia: OpenStreetMap. https://en.wikipedia.org/wiki/OpenStreetMap. Accessed 18 Feb 2019
19. Xu, M., Wu, J., Liu, M., Xiao, Y., Wang, H., Hu, D.: Discovery of critical nodes in road networks through mining from vehicle trajectories. IEEE Trans. Intell. Transp. Syst. **20**, 583–593 (2018)
20. Yin, H., Wolfson, O.: A weight-based map matching method in moving objects databases. In: Proceedings of 16th International Conference on Scientific and Statistical Database Management, pp. 437–438. IEEE (2004)
21. Yuan, J., Zheng, Y., Zhang, C., Xie, X., Sun, G.Z.: An interactive-voting based map matching algorithm. In: Proceedings of the 2010 Eleventh International Conference on Mobile Data Management, pp. 43–52. IEEE Computer Society (2010)