An Advanced Interactive-Voting Based Map Matching Algorithm for Low-sampling-rate GPS data

Yaying Zhang
Key Laboratory of Embeded Sytems and Service Computing
Tongji University
Shanghai, China
yaying.zhang@tongji.edu.cn

Yulong He
Key Laboratory of Embeded Sytems and Service Computing
Tongji University
Shanghai, China
yulong_he@tongji.edu.cn

***Abstract*—The availability of GPS data from in-vehicle devices has greatly enriched the location based system applications, in which map matching plays an important role. For the low-sampling-rate GPS data, existing map matching algorithms would have many challenges such as the high error rate and inefficiency on the complex urban road network. This paper presents an advanced map matching algorithm based on interactive-voting for low-sampling-rate data of taxi GPS trajectories. The algorithm employs spatial analysis function, temporal analysis function, and road analysis function with two constraints to measure the relationship between consecutive candidate points in map-matching. The constraints in our algorithm not only reduce much computation but also significantly improve the accuracy of the algorithm. Based on the interaction between GPS sampling points, we propose a novel voting method called *candidate edge voting* to get the best map matching result. We evaluate our algorithm using a one-month real-world trajectory dataset. The proposed algorithm outperforms in both accuracy and efficiency.**

***Index Terms*—map matching, GPS, road network, voting**

I. Introduction

With the widespread use of GPS-equipped devices, cell phones, vehicles, unmanned aerial vehicles, and even the watches can collect their location data. A rich data pool of GPS trajectories is gradually becoming the data source for intelligent transportation systems and related services [1], [2]. In most big cities, there is a significant number of taxies driving in the urban with GPS devices, which contributes to the tremendous amount of taxi trajectories. Subject to the factors such as electromagnetic interference and power consumption, GPS data from running taxies with passengers will always be recorded in an interval that is over one minute. As taxies may have driven quite a long distance in a minute, GPS position may not be available all the time along the road segment. Practically it is quite probable that most of the data have low-sampling-rate (the sampling interval is larger than 60 seconds) with deviation. All of these data would need post-processing, e.g. map matching. The purpose of a map-matching algorithm is to align the GPS point to the road segment in the road network [3].

978-1-5386-5053-0/18/$31.00 2018 IEEE

Over the last two decades, there were abundant researchers dealing with taxi trajectories [4]–[7] and map matching algorithms [8], [9] on GPS data. In terms of the use of additional information, map matching algorithms can be divided into four categories: geometric [10], [11], topological [12], [13], probabilistic [14] and other advanced techniques [15], [16]. Most of them focused on high-sampling-rate GPS trajectories. Map matching for low-sampling-rate GPS trajectories is more difficult, especially in the complex road network in urban cities. Mohammed Quddus et al. [17] proposed a method in map matching low-sampling-rate GPS data using the shortest path and vehicle trajectory, in which vehicle heading was used as critical data. However, practically, high-precision GPS devices are not always onboard on taxies due to their cost. The sampled vehicle heading is still unreliable.

The well-known low-sampling-rate GPS data matching algorithms are the ST-Matching algorithm [18] and the IVMM algorithm [19] proposed by Zheng Y et al. These algorithms consider the spatial and temporal information of a GPS trajectory and model the weighted mutual influences between GPS points. However, the process of the algorithms is complicated, and the data must be map matched repeatedly. Moreover, these algorithms still can be further improved in terms of the following aspects:

(1) The simple summation of weights on the path would have the consequence that if the cumulative weight of the wrong candidate path is much larger than the truly matched path in the first few candidate points, the final cumulative weight of the wrong path will be greater than the correct path even if the weight of the wrong path in its subsequent candidate is comparatively small.

(2) Without considering the practically resaonable range of vehicles' speed, there are occasions that the distance between two sampling points of the matching trajectory is too long for any vehicle to travel during the specified time. Furthermore, all GPS sampling points are considered to be effective, which would not work well for the trajectory that has the GPS

sampling point with offset more than 100m. Those sample points should be removed in data post-processing.

(3) The result of some trajectories after map-matching would be in "zigzag" especially on the urban elevated road (i.e., the vehicle travels up and down frequently on the elevated road).

To solve the above-mentioned problems in the map matching low-sampling-rate GPS trajectory data, we propose an advanced map matching algorithm based on interactive-voting. Originated from IVMM [19], our algorithm employs spatial analysis function, temporal analysis function, and road analysis function with two constraints to measure the relationship between consecutive candidate points in map-matching.

The remainder of this paper is as follows. Section II states the problem of map matching algorithm and clarifies the preliminary ideas. The detail of the proposed algorithm is presented in Section III. The experiment results are demonstrated and analyzed in Section IV. The conclusion is given in Section V.

II. Preliminary

A. Map Matching Problem

Firstly we give some definitions related to the map matching algorithm.

- *(Road Segment)* A road segment r as a basic unit in a road network is a directed (one-way or bidirectional) edge. Each road segment has a length r.length and its speed limit r.v. Every road segment has its start point r.s and its end point r.e. A road intersection c is a connected point of two or more road segments.
- *(GPS Trajectory)* A GPS trajectory T is a sequence of GPS sampling points $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$. Each GPS point has key information of latitude, longitude, timestamp, direction and speed. The time interval between any consecutive GPS points more than 60 seconds is defined as low-sampling-rate.
- *(Road Network)* A directed graph G(V, E), where V is a set of vertices representing road intersections or the ends of the road segment, E is a set of edges representing the road segments.
- *(Path)* A Path P is a road segment sequence $r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_n$, where $r_{i-1}.e = r_i.s,\ (1 \leq i \leq n)$. r.s and r.e represent the start point and end point of the road segment r respectively.
- *(Local Optimal Path)* For each candidate point c_i^j of sampling point p_i in a GPS trajectory T, it is assumed that the optimal path through the candidate point c_i^j is the real path of T.

Map Matching: Given a GPS trajectory T and a road network G, find a real path P of T in G.

B. Candidate Points Selection

The candidate points to the sampling point are critical for map matching. For every GPS point $p_i, i = 1, 2, \ldots, n$, in the trajectory, there are several candidate matching points $c_i^j, j = 1, 2, \ldots, m$. We use the KNN algorithm proposed in [20] to obtain candidate points. Even the road network is complex or the number of GPS data is very large, this algorithm can still find the candidate points quickly. In Fig. 1, the sampling point p_2 is projected vertically in the projection area, obtain the candidate points c_2^1 and c_2^2. If there is a crossroad in the projection area, the intersection of the road should also be considered as a candidate point, i.e. the candidate point c_2^3 is obtained. Considering the intersection as a candidate point is critical. If the sampling point p_1 and p_3 only have one candidate point c_1^1 and c_3^1 respectively, point p_2 obtains the candidate points c_2^1 and c_2^2 by the vertical projection and does not consider the intersection c_2^3 as a candidate point, neither path $c_1^1 \rightarrow c_2^1 \rightarrow c_3^1$ nor $c_1^1 \rightarrow c_2^2 \rightarrow c_3^1$ could be selected in the map matching. In this scenario, most of the vehicles wouldl not choose to have u-turns.

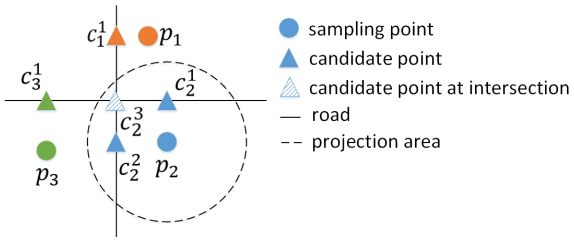


Fig. 1. Sampling point and candidate points selection

C. Mutual Influence of Candidate Points

The GPS sampling points are influential with each other, especially for adjacent ones. In Fig. 1, we have confidence that the final path from p_1 to p_3 goes via c_2^3, not c_2^1 or c_2^2. Because the sampling point p_2 is influenced by both p_1 and p_3. Apparently, the closer to the point, the greater the influence. Based on the mutual influence of sampling points and the voting process [19], we propose an Advanced Interactive-Voting Based Map Matching Algorithm (AIVMM). For each candidate point of the sampling point, find the local optimal path which passes through the candidate point by *Position Context Analysis* process discussed in Section III.

D. Find Final Matching Path

A basic assumption behind our work is that the local optimal paths can reflect the possibility of the path which the final matching path is passing through. If path $c_{i-1}^t \rightarrow c_i^s$ (denoted by P) is not passed by any local optimal path, the final matching path will not include the path P. With this assumption, find the path from candidate of p_{i-1} to candidate of p_i which is most heavily traversed by all local optimal paths and defined the path as a final subpath. Moreover, we propose a process called candidate edge voting to count how many local optimal paths would pass through the candidate path. For each local optimal path, if the two consecutive candidate points are on the path, the vote of the path from one of the two candidate points to the other is increased by one. Finally, all the final subpaths can be obtained by the candidate edge voting process. Moreover, splice them together, a final matching path will be formed.

III. Advanced Interactive-Voting Based Map Matching Algorithm

GPS sampling points have information about its latitude, longitude and time stamp. There are also vehicles' instantaneous direction and instant speed. The information accuracy is determined by the precision of GPS devices. Inaccurate information could cause a high interference. For the instantaneous direction and speed in the sampled GPS data, which is not reliable enough, we will not simply use their value directly but extract useful information from the relationship between different data points, such as the average speed between two points, the weight between the two points.

The proposed AIVMM algorithm consists of four phases: candidate preparation, position context analysis, mutual influence modeling, and interactive voting. Fig. 2 shows a framework of the AIVMM algorithm.

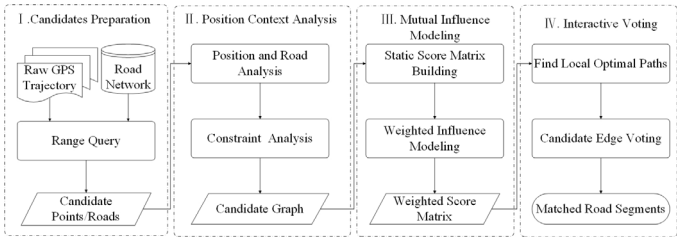


Fig. 2. Overview of the AIVMM algorithm

We use the range query of KNN algorithm based on KD tree [20] to obtain the candidate points for each sampling point in the first phase as described in Section II-B. Then in the second phase, we use the position and road analysis and constraint analysis to construct the candidate graph. After building weighted score matrix by a static score matrix with weighted influence modelling in the third phase, we can find the local optimal paths. Based on the paths, all the candidate edges are voted for the optimal matching paths in the last phase.

A. Position And Road Analysis

We use the *Spatial Temporal Analysis* to measure the influence between two consecutive candidate points. The measurement error of GPS points satisfies the Gaussian distribution $N(\mu, \sigma^2)$ [21], so the observation probability is as:

$$N(c_i^j) = e^{-\frac{(x_i^j-\mu)^2}{2\sigma^2}} . \quad (1)$$

where c_i^j is a candidate point of sampling point p_i and x_i^j is the Euclid distance from candidate c_i^j to the sampling point.

The *spatial analysis function* [18] from candidate point c_{i-1}^t to candidate point c_i^s is defined as:

$$F_s(c_{i-1}^t \to c_i^s) = N(c_i^s) \times V(c_{i-1}^t \to c_i^s),\ 2 \le i \le n . \quad (2)$$

where $V(c_{i-1}^t \to c_i^s)$ is the transition probability. It aims to measure the similarity of shortest path and the straight path between the two consecutive candidate points. It is defined as:

$$V(c_{i-1}^t \to c_i^s) = \frac{d_{i-1 \to i}}{w_{(i-1,t)\to(i,s)}} . \quad (3)$$

where $d_{i-1\to i}$ is the Euclidian distance from sampling point p_{i-1} to sampling point p_i, and $w_{(i-1,t)\to(i,s)}$ is the length of the shortest path from candidate c_{i-1}^t to c_i^s. The *temporal analysis function* is defined as:

$$F_t(c_{i-1}^t \to c_i^s) = \frac{\hat{v}_{(i-1,t)\to(i,s)}}{\left|\hat{v}_{(i-1,t)\to(i,s)} - \bar{v}_{(i-1,t)\to(i,s)}\right| + \hat{v}_{(i-1,t)\to(i,s)}} . \quad (4)$$

where $\hat{v}_{(i-1,t)\to(i,s)}$ is the weighted speed limit of the shortest path from candidate point c_{i-1}^t to c_i^s, and $\bar{v}_{(i-1,t)\to(i,s)}$ is the average speed of the vehicle traveling along the shortest path between candidate points c_{i-1}^t to c_i^s. When there is no traffic accident, the vehicle speed is generally close to the road speed limit. Therefore, a suitable *temporal analysis function* should get a maximum value near the road speed limit.

In many map matching algorithms, we often see in the complex urban road network with the elevated road, the result of the matching is that the vehicle travels up and down the elevated road. As shown in Fig. 3, $\hat{ab}$ is the elevated road and $\hat{cd}$ is the surface road. $\hat{ef}$ and $\hat{gh}$ is the off-ramp and on-ramp respectively. The long dashed line is the matching result by IVMM algorithm, and we can see the vehicle drove down from the elevated road and then drove up to the elevated road. In actual driving, vehicles tend to be along the current road, not travels up and down the elevated road.

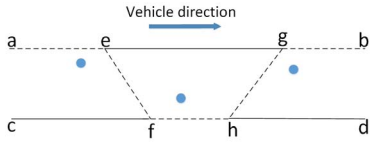


Fig. 3. Vehicle travels up and down the elevated road

In the map, the road is often divided into several levels, and each level has a different speed limit of the road. The level of the elevated road is higher than the surface road, that is, the speed limit of the elevated road is greater than the surface road. Therefore, we use road level factor to model the vehicle's tendency to be stay on the road with high speed limit. The *road level factor* (RLF) from $c_{i-1}^t \to c_i^s$ is defined as:

$$RLF(c_{i-1}^t \to c_i^s) = \frac{v_s}{(v_d - v_s) + v_s} . \quad (5)$$

where v_s and v_d represent the speed limit of the road where c_{i-1}^t and c_i^s are located. Thus, the weight function for path $c_{i-1}^t \to c_i^s$ could be extendeded as:

$$F(c_{i-1}^t \to c_i^s) = F_s(c_{i-1}^t \to c_i^s) \times F_t(c_{i-1}^t \to c_i^s) \times RLF(c_{i-1}^t \to c_i^s) . \quad (6)$$

B. Constraint Analysis

According to our statistical analysis on the values of weight function collected from extensive experiments, the weight $F(c_{i-1}^t \to c_i^s)$ of wrong matching path $c_{i-1}^t \to c_i^s$ is always very low (usually less than 1.0×10^{-5}) and none of the weight of the correct matching path is less than 1.0×10^{-5}. So we set

a threshold θ to the weights between the two candidate points as a Constraint. It is defined as:

Constraint 1: When $F(c_{i-1}^t \to c_i^s) < \theta$, where θ is a user-specified threshold, path $c_{i-1}^t \to c_i^s$ is considered as an error matching.

There are two advantages of Constraint 1. One is to eliminate the disadvantage of simple weights summation mentioned in the Section I. Based on the Position and Road Analysis and Constraint 1, we can eliminate most of the error paths. So even the cumulative weight of the wrong candidate path is much larger than the truly matched path in the first few candidate points, we will not choose the wrong path in its subsequent candidate. The greatest final cumulative weight will be the truly matched path. And the other advantage is that it eliminates those paths with too small weights, reducing the computational effort in finding the local optimal path.

Only using the *temporal analysis function* to measure the vehicle speed is insufficient because it does not consider whether the speed of the vehicle is within a reasonable range when travelling along the matching path. Road speed limit is critical information. From the map we can get the speed limit of each road and calculate the weighted average speed limit $\bar{v}_\omega(c_{i-1}^t \to c_i^s)$ for all paths where $c_{i-1}^t \to c_i^s$ passes. Considering the relationship between the average speed of vehicle and the weighted average speed limit of roads, the speed of vehicle on the matching path would not exceed a reasonable range.

Constraint 2: $\bar{v}(c_{i-1}^t \to c_i^s)$ is the average speed of the vehicle traveling along the shortest path between candidate points c_{i-1}^t to c_i^s. $\bar{v}_\omega(c_{i-1}^t \to c_i^s)$ is the weighted average speed limit for all paths which $c_{i-1}^t \to c_i^s$ passes. When $\bar{v}(c_{i-1}^t \to c_i^s) > \alpha\bar{v}_\omega(c_{i-1}^t \to c_i^s)$, $1 \le \alpha \le 2$, path $c_{i-1}^t \to c_i^s$ is considered as an error matching.

GPS-enabled devices in taxies are typically accurate to within 20 meters. Their accuracy worsens as the vehicles are close to buildings, bridges, and trees. Nevertheless, the offset of most GPS points is no more than 100 meters. However, there are still a few GPS sampling points collected which are out of this range. In this case, such points should be removed to prevent their inteference. Lee and Krumm [23] used the Kalman and particle filters to fix the noisy trajectory points. But the method cannot be applied to the low-sampling-rate GPS trajectory. Since the Kalman filter will lose the tracking target when the moving target is blocked for a long time.

We present a method to filter noisy trajectory points based on Constraint 1 and Constraint 2. When all candidate paths between sampling points p_{i-1} and p_i do not satisfy the Constraint 1 and Constraint 2, it is considered that sampling point p_i is noisy point. The sampling point p_i is removed and the successor point of p_{i-1} is changed to p_{i+1}. Then the map matching process is ressumed.

After position and road analysis and constraint analysis, a candidate graph is constructed. The nodes are the set of candidate points, and the edges are set of shortest paths between two adjacent candidate points. The nodes and edges are all assigned weight values based on the results of position and road analysis.

C. Mutual Influence Modeling

Before we compute the weight of mutual influence, we build a *Static Score Matrix* [19] as $M = diag\{M^{(2)}, M^{(3)}, \cdots, M^{(n)}\}$, where $M^{(i)} = (m_{ts}^{(i)})_{a_{i-1}\times a_i} = (F(c_{i-1}^t \to c_i^s))_{a_{i-1}\times a_i}$. Each item in this static score matrix represents the weight of a candidate point only considering the information of two consecutive points. This information does not reflect the interactive mutual influence. So we can build it as a static matrix to reduce computation when we compute the weight of mutual influence.

To model the weighted influence of the candidate points, an (n - 1)-dimension *Distance Weight Matrix* [19] is defined as below:

$$W_i = diag\{w_i^{(1)}, w_i^{(2)}, \cdots, w_i^{(i-1)}, w_i^{(i+1)}, \cdots, w_i^{(n)}\}\ . \quad (7)$$

where $w_i^{(j)} = e^{\frac{(dist(p_i,p_j))^2}{\beta^2}}$ and $dist(p_i, p_j)$ is the Euclidian distance between p_i and p_j, β is a parameter with respect to the road network. This matrix gives weights for the effect with distance of all other points to p_i. For $i = 2, 3, \cdots, n$, the weight of $c_{i-1}^t \to c_i^s$ with distance influence can be calculated as:

$$\Phi_i^{(j)} = (\varphi_{ts}^{(i,j)})_{a_{i-1}\times a_i} = \begin{cases} w_i^{j-1}M^{(j)} & \text{if } 1 \le i \le j \\ w_i^j M^{(j)} & \text{otherwise} \end{cases} \quad (8)$$

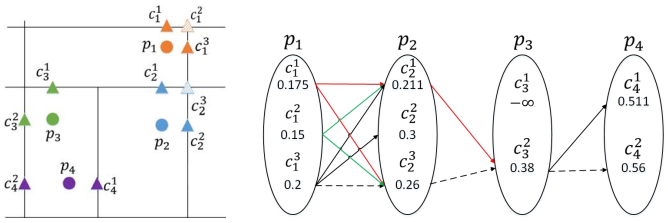
the weighted score matrix Φ_3 is like:

$$\Phi_3 = \begin{bmatrix} 0.4 & 0.3 & -\infty & -\infty \\ 0.35 & 0.25 & -\infty & -\infty \\ -\infty & -\infty & 0.075 & 0.175 \\ -\infty & -\infty & 0.05 & 0.1 \end{bmatrix}$$

Matrix Φ_3 is the weighted score matrix for sampling point p_3. And the $\Phi_3^{(1,1)} = 0.4$ represents the weight of $c_1^1 \to c_2^1$ is 0.4.

D. Find Local Optimal Path

For each candidate point $c_i^j, i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$, suppose c_i^j is the correct point in the final map matching result. Find a path with the greatest probability of passing through the point c_i^j as a local optimal path. We use the GPS trajectory in Fig. 4(a) to illustrate the process of find the local optimal path. In Fig. 4(b), the value represents the cumulative weight of the candidate point from p_1. The solid line and dotted line represents that there is a legitimate path from c_{i-1}^t to c_i^s (i.e., any vehicle can travel through the path $c_{i-1}^t \to c_i^s$ in reasonable time). In this example, we assume c_3^2 is the correct point and set the weight of path $c_2^1 \to c_3^1$ and $c_2^2 \to c_3^1$ are $-\infty$. And this implies that the path must pass through the candidate point c_3^2. The cumulative weight $fValue(c_i^s) = max\{fValue(c_{i-1}^t) + \varphi_{ts}^{(i,j)}\}$, $t = 1, 2, \cdots, n$. $\varphi_{ts}^{(i,j)}$ is the weight of $c_{i-1}^t \to c_i^s$ with the distance influence of p_i and p_j. After calculating the cumulative weight, we obtain the fValues of all p_4's candidate points. From p_4 we



(a) Original map (b) Find the local optimal path

Fig. 4. Find the local optimal path

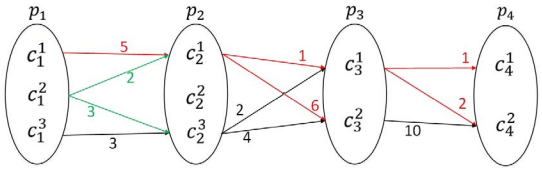


Fig. 5. Vote of Candidates

can obtain that th e local optimal candidate point is c_4^2 because $fValue(c_4^2) > fValue(c_4^1)$. And from c_4^2 we can obtain that the previous local optimal candidate point c_3^2. Likewise, we obtain the local optimal path is $c_1^3 \rightarrow c_2^3 \rightarrow c_3^2 \rightarrow c_4^2$ as shown by the dotted line in Fig. 4(b).

E. Candidate Edge Voting

After finding local optimal path of each candidate point c_i^j, we get a set of local optimal paths. With the assumptions that if most of the local optimal paths pass through a road and then the final map matching result is also pass through the road, we could obtain the final map matching results. Based on the set of local optimal paths, we can get the path with the greatest probability of the final map matching result. Different from the interactive voting method in [19]which concentrates on the voting of candidate point, we propose a novel *candidate edge voting* method to get the final map matching result. A candidate point passed by local optimal paths more frequently may not be the point in the true path.

In Fig. 5, we continue to use the GPS trajectory in Fig. 4(a) to illustrate the *candidate edge voting* process. It should be noted that the *edge* here refer to the corresponding relationship $c_{i-1}^t \rightarrow c_i^s$ of the consecutive sampling points p_{i-1} and p_i's candidate, which are presented in Fig. 5 as edges. After finding the local optimal path, if the candiate point c_{i-1}^t and c_i^s is in the path P, the vote of path $c_{i-1}^t \rightarrow c_i^s$ is increased by one. In Fig. 5, the value of line represents the number of local optimal path pass through the roads in the set of optimal paths. For $p_1 \rightarrow p_2$, the number of vote edges $c_1^1 \rightarrow c_2^1$, $c_1^2 \rightarrow c_2^1$, $c_1^2 \rightarrow c_2^3$ and $c_1^3 \rightarrow c_2^3$ is 5, 2, 3 and 3 respectively. The largest vote edge is $c_1^1 \rightarrow c_2^1$. Likewise, the largest vote edge for $p_2 \rightarrow p_3$ is $c_2^1 \rightarrow c_3^2$ and for $p_3 \rightarrow p_4$ is $c_3^2 \rightarrow c_4^2$. Finally, we get the final matching result is $c_1^1 \rightarrow c_2^1 \rightarrow c_3^2 \rightarrow c_4^2$. The process of candidate edge voting is described in Algorithm 1.

Algorithm 1 Candidate Edge Voting

Input: The set of local optimal paths LOP
Output: final map matching result P: $c_1^{s_1} \rightarrow \cdots \rightarrow c_i^k \rightarrow \cdots \rightarrow c_n^{s_n}$

Let $vote[][][]$ denote the number of vote edge computed so far
Let $p[]$ denote the final map matching result of candidate point
for $i = 1 \rightarrow n$ **do**
  for $j = 1 \rightarrow m$ **do**
    $c_{from} = LOP[i][j]$
    $c_{to} = LOP[i][j+1]$
    $vote[j][c_{from}][c_{to}] = vote[j][c_{from}][c_{to}] + 1$
  end for
end for
$p[1], p[2] = argmax_{c_{from}, c_{to}}\{vote[0][c_{from}][c_{to}], c_{from} = 1, 2, \ldots, a_1.\ c_{to} = 1, 2, \ldots, a_2\}$ /*find the final map matching candidates of p_1 and p_2, a_i is the number of p_i's candidate points*/
for $s = 2 \rightarrow m$ **do**
  $p[s] = argmax_{c_{to}}\{vote[s][p[s]][c_{to}], c_{to} = 1, 2, \ldots, a_{s+1}\}$
end for
return p

IV. Evaluation

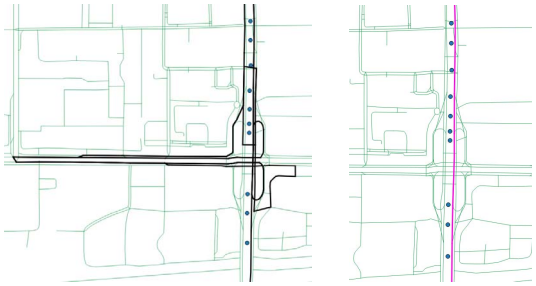
We use real-world data to evaluate the proposed algorithm. First, we will introduce the setting of the evaluation and then analyse the evaluation result.

A. Data

We use the road network of Beijing, which has 144,151 road segments and 108,379 intersections. The GPS data was collected by 28,203 taxies in Beijing from October 30, 2015, to November 30, 2015, containing 1,821,060 trajectories. Among the trajectories, there is a large number of GPS sampling points gathered due to vehicle stoppage. So we remove those points and only use the GPS sampling points when the taxi has passengers. We have 544,659,447 sampling points in the GPS data. In our GPS dataset, 53% of the consecutive two sampling points' time interval is 60-69s. More than 60% of the sampling points are low-sampling-rate (the sampling interval is more than 60 seconds). To evaluate the performance of the AIVMM algorithm in this paper, we randomly select 100 GPS trajectories from the above-refined dataset, most of the data are low-sampling-rate. The time interval between consecutive sampling points of the data is mainly concentrated in the 30 seconds to 120 seconds. According to the original GPS position and personal experience, the 100 sampled trajectories have been manually map-matched to get a ground truth.

B. Evaluation Setup

We set $k = 10$ as the maximum number of candidate points for each sample point, and the query radius is $r = 100m$. In [22], Chenguang introduced that the GPS offset accords with



(a) IVMM (b) AIVMM

Fig. 6. A screenshot of maching result about simple weight sum

the normal distribution and the probability of the GPS offset within the range $\pm 2\sigma$ is 95.44%. So the parameters of normal distribution function is $\mu = 5$ and $\sigma = 25$. The distance weight parameter is $\beta = 5km$. The algorithms are implemented in C++, on an Intel i5-2400 PC with 4GB memory on Linux Mint 18 operating system.

C. Evaluation Approach

The accuracy is the correctness of matching the road, and the efficiency is to compare the running time of the algorithms on the same platform. We use the correct matching percentage (CMP) to evaluate the accuracy of the algorithms.

$$CMP = \frac{|Correct\ matched\ road\ segments|}{|Road\ segments\ to\ be\ matched|} \times 100\% \quad (9)$$

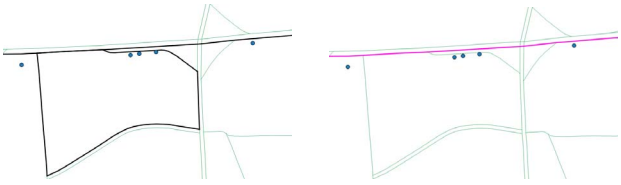
D. Result and Analysis

Fig. 6-9 present the visualized matching result of AIVMM algorithm compared with IVMM algorithm using the same GPS trajectory. In all these pictures, the dots represent GPS sampling points, the green line represents road network, and the black and pink lines represent matching results. The simple summation of weights along the path in IVMM algorithm will decrease the accuracy. Although the speed of the vehicle is within a reasonable range when travelling along the matching path in Fig. 6(a), the matching result is not correct. The AIVMM algorithm can obtain the correct results as shown in Fig. 6(b).

When there is no constraint on the vehicle speed limit, after several consecutive GPS sampling points with large offsets, the matching result of IVMM algorithm is an extended circle as shown in Fig. 7(a). Adding the vehicle speed limit in the AIVMM algorithm, the matching result is a straight line as shown in Fig. 7(b).

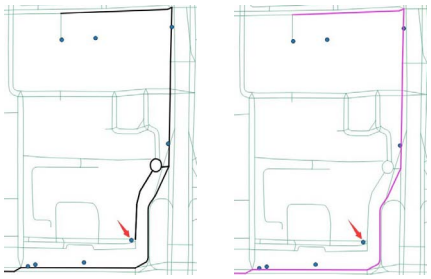
In Fig. 8, the offset of the point indicated by the arrow is greater than 100 meters (i.e., the sampling point is a noisy point). AIVMM algorithm can find the noisy point and remove it. So we can obtain correct path as shown in Fig. 8(b).

Fig. 9 presents the matching result on the complex urban road network. The result of IVMM algorithm is shown in Fig. 9(a). The vehicle travels down and up the elevated road. The matching result of AIVMM algorithm is a straight line as shown in Fig. 9(b).



(a) IVMM (b) AIVMM

Fig. 7. A screenshot of maching result about vehicle speed



(a) IVMM (b) AIVMM

Fig. 8. Screenshot of maching result with noisy point

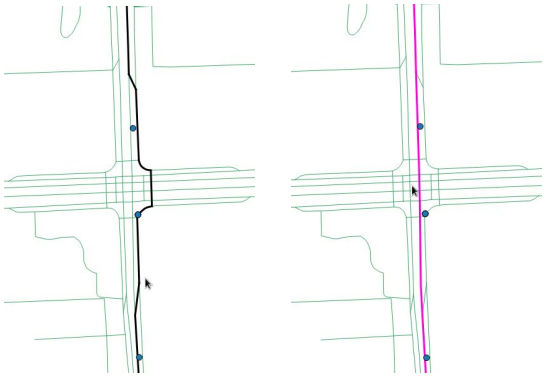
Fig. 10(a) shows the result of the matching accuracy comparison based on correct matching percentage. The accuracy of AIVMM algorithm is nearly 90%, about 15% and 58% better than the IVMM and ST-Matching algorithm respectively. The accuracy of ST-Matching algorithm is low because it does not take into account the influence relationship between sampling points influence relationship between sampling points.

We have noticed that regardless of the vehicle's speed, the IVMM algorithm can get correct map matching results in the sparse road network. But in dense road network it will have wrong result. The ST-Matching algorithm could not get a good performance in either of these two cases. AIVMM algorithm improves the accuracy by removing a large number of wrong candidate paths based on the two constraints. Although the AVIMM algorithm do have good performance in many cases, it still have space to be improved. When the vehicle speed is too slow (less than $15km/h$) or too fast, the Constraint 2 can not be applied to this situation, which will lead to the incorrect matching result.

Fig. 10(b) shows the running time of IVMM, AIVMM and ST-Matching algorithms. The 100 GPS trajectories for test are divided into ten groups. Each group has ten GPS trajectories. We compare the running time of the algorithm on the same platform. We can see that the AIVMM algorithm is about faster 5 to 10 times than the IVMM algorithm. The efficiency of the AIVMM algorithm is almost the same as that of the ST-Matching algorithm, but the accuracy is much higher.

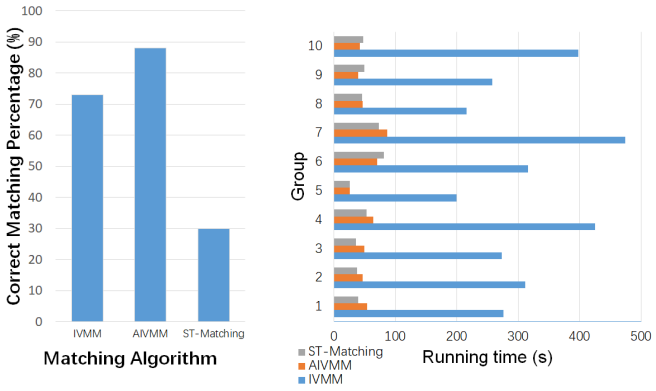
V. Conclusion

In this paper, we investigate the map matching algorithm for low-sampling-rate GPS sampling points. Based on the sampling points mutual influence, we present a new map matching method that uses the spatial analysis function, temporal analysis function, and road analysis function to measure the



(a) IVMM (b) AIVMM

Fig. 9. Screenshot of maching result with urban elevated road



(a) Accruracy (b) Efficiency

Fig. 10. Performance Analysis (IVMM V.S. AIVMM)

relationship between consecutive candidate points. Improved accuracy and efficiency compared to IVMM algorithm is achieved by introducing the two constraints and filtering noisy trajectory points. Moreover, a novel candidate edge voting method is proposed to find the reliable map matching path. Experimental results have shown that the correct matching percentage of AIVMM algorithm is much higher than IVMM and ST-Matching algorithm. Even in the dense road network, the proposed method has a good matching result.

ACKNOWLEDGMENT

This work is partly supported by the Fundmental Research Funds for the Central Universities(22120170224).

REFERENCES

[1] Tang, Jinjin, et al. "Estimating the most likely spacetime paths, dwell times and path uncertainties from vehicle trajectory data: A time geographic method." Transportation Research Part C Emerging Technologies, 2016, pp.17–194.
[2] Westgate, Bradford S., et al. "Large-network travel time distribution estimation for ambulances." European Journal of Operational Research, 2016, pp.322–333.
[3] Quddus, Mohammed A., et al. "A general map matching algorithm for transport telematics applications." Gps Solutions, 2003, pp.157–167.
[4] Hunter, Timothy, et al. "Path and Travel Time Inference from GPS Probe Vehicle Data.", 2009.
[5] Yuan, Jing, et al. "T-drive: driving directions based on taxi trajectories." Sigspatial International Conference on Advances in Geographic Information Systems ACM, 2010, pp.99–108.
[6] Herring, Ryan, et al. "Estimating arterial traffic conditions using sparse probe data." International IEEE Conference on Intelligent Transportation Systems IEEE, 2010, pp.929–936.
[7] Zhang, Jindong, et al. "Efficient vehicles path planning algorithm based on taxi GPS big data." Optik - International Journal for Light and Electron Optics, 2016, pp.2579–2585.
[8] Quddus, Mohammed, and S. Washington. "Shortest path and vehicle trajectory aided map-matching for low frequency GPS data." Transportation Research Part C Emerging Technologies, 2015, pp.328–339.
[9] Koller, H., Widhalm, P., Dragaschnig, M., Graser, A.: Fast Hidden Markov Model Map-Matching for Sparse and Noisy Trajectories. IEEE, International Conference on Intelligent Transportation Systems (pp.2557-2561). IEEE. (2015)
[10] Bernstein, D., Kornhauser, A.: An introduction to map matching for personal navigation assistants. (1998)
[11] Greenfeld, Joshua S. "Matching GPS Observations to Locations on a Digital Map." Transportation Research Board 81st Annual Meeting, 2002.
[12] Velaga, Nagendra R., M. A. Quddus, and A. L. Bristow. "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems." Transportation Research Part C Emerging Technologies, 2009, pp.672-683.
[13] Yin, Huabei, and O. Wolfson. "A Weight-based Map Matching Method in Moving Objects Databases." International Conference on Scientific and Statistical Database Management, 2004, pp.437–438.
[14] Pink, O., Hummel, B.: A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. In: 2008 11th International IEEE Conference on Intelligent Transportation Systems (pp. 862–867). IEEE. (2008)
[15] Newson, P., Krumm, J.: Hidden Markov map matching through noise and sparse-ness. In: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems (pp. 336-343). ACM. (2009)
[16] Wang, B., Yang, Y., Sun, Y. "Research on high accurate and real-time map-matching algorithm in LBS." GMC, 2006, pp.258–262.
[17] Quddus, Mohammed, and S. Washington. "Shortest path and vehicle trajectory aided map-matching for low frequency GPS data." Transportation Research Part C Emerging Technologies, 2015, pp.328–339.
[18] Lou, Yin, et al. "Map-matching for low-sampling-rate GPS trajectories." ACM Sigspatial International Symposium on Advances in Geographic Information Systems, Acm-Gis 2009, November 4-6, 2009, Seattle, Washington, Usa, Proceedings DBLP, 2009, 352–361.
[19] Yuan, Jing, et al. "An Interactive-Voting Based Map Matching Algorithm." Eleventh International Conference on Mobile Data Management IEEE, 2010, pp.43–52.
[20] Zhang, Guangyi, and F. Li. "Application of the KNN algorithm based on KD tree in intelligent transportation system.", 2014, pp.832–835.
[21] Pfoser, Dieter, and C. S. Jensen. "Capturing the Uncertainty of Moving-Object Representations." International Symposium on Advances in Spatial Databases Springer-Verlag, 1999, pp.111–132.
[22] Chen-Guang, L. U. "GPS Information and Rate-Tolerance and its Relationships with Rate Distortion and Complexity Distortions." Journal of Chengdu University of Information Technology, 2012.
[23] Lee, Wang Chien, and J. Krumm. "Trajectory Preprocessing.", 2011, pp.3–33.