An Interactive-Voting Based Map Matching Algorithm

Jing Yuan*
University of Science and Technology of China
Hefei, China
yuanjing@mail.ustc.edu.cn

Yu Zheng
Microsoft Research Asia
Beijing, China
yuzheng@microsoft.com

Chengyang Zhang
University of North Texas
Denton, TX. U.S.A.
zhchengyang@gmail.com

Xing Xie
Microsoft Research Asia
Beijing, China
xingx@microsoft.com

Guang-Zhong Sun
University of Science and Technology of China
Hefei, China,
gzsun@ustc.edu.cn

***Abstract*—Matching a raw GPS trajectory to roads on a digital map is often referred to as the Map Matching problem. However, the occurrence of the low-sampling-rate trajectories (e.g. one point per 2 minutes) has brought lots of challenges to existing map matching algorithms. To address this problem, we propose an Interactive Voting-based Map Matching (IVMM) algorithm based on the following three insights: 1) The position context of a GPS point as well as the topological information of road networks, 2) the mutual influence between GPS points (i.e., the matching result of a point references the positions of its neighbors; in turn, when matching its neighbors, the position of this point will also be referenced), and 3) the strength of the mutual influence weighted by the distance between GPS points (i.e., the farther distance is the weaker influence exists). In this approach, we do not only consider the spatial and temporal information of a GPS trajectory but also devise a voting-based strategy to model the weighted mutual influences between GPS points. We evaluate our IVMM algorithm based on a user-labeled real trajectory dataset. As a result, the IVMM algorithm outperforms the related method (ST-Matching algorithm).**

Keywords-map matching; GPS trajectory; road network; voting

I. INTRODUCTION

The increasing popularity of GPS-enabled device has facilitated users to track moving objects, such as vehicles and people. However, as the readings of a GPS sensor have positioning errors and sampling errors [4], the departure of the GPS tracking data from the actual trajectory can hardly be avoided. To match an original GPS tracking data to a digital map or a digital road network is often referred to as **Map Matching**. The general purpose of a map matching algorithm is to identify the true road segment on which a user (or a vehicle) is/was travelling. Map matching is a key procedure in many location-based applications, such as vehicle navigation [8], fleet management [9], intelligent transport systems (ITS) [17], and many other location based services [2, 7]. Since most of the civilian GPS devices and GPS modules in smart phones are low-end and low accuracy, a sophisticated and reliable map matching algorithm is crucial for these location-based services.

There has been an increasing attention on the map matching problem. The majority of the existing map matching algorithms consider the scenario that the sampling rate is high (e.g. one sample per 10 seconds). However, in practice, there exists a large quantity of low-sampling-rate GPS tracking data (the sampling interval is more than 2 minutes). For instance, to save the communication and energy cost, the taxis usually report their GPS positions to the dispatching center with low-sampling-rate data. Fig. 1 presents the statistical distribution of the sampling intervals of the GPS trajectories generated by 10,000+ taxies in Beijing in a week. The average time interval of the data set is 3.27 minutes. According to the result in the chart, only 34% of the data is high-sampling-rate (the sampling interval is less than 1 minute) data. More than 60% of the data is low-sampling-rate data.

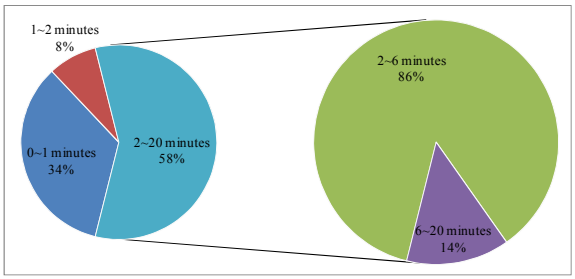


Figure 1. Distribution of the sampling interval

This poses new challenge for conventional map matching algorithms since as the sampling rate grows, the details between two sampling points are missing. For example, if the sampling interval is 3 minutes, the distance between two sampling points may reach to 2000 meters even if a vehicle's speed is 40 km/h. In a dense urban road network, a taxi may pass through several road segments during this period. Another problem is that the matched road segments of most exiting methods may be disconnected when the GPS sampling interval is large. Therefore, an adaptive and robust map matching algorithm with acceptable accurate is needed for this scenario.

In this paper, we investigate the problem of map matching low-sampling-rate GPS tracking data. To match this kind of data to the real map, we need to make full use of the interior information of the tracking data as well as the topology of the road networks. For instance, Fig. 2 shows a real road map and the GPS tracking data (the red points). We can easily manual

*This work was done when Jing Yuan was an intern at Microsoft Research Asia.

978-0-7695-4048-1/10 $26.00 © 2010 IEEE
DOI 10.1109/MDM.2010.14

match the trajectory to the blue dot line which is the true path of this vehicle. This procedure indicates our key insights:

1) Position Context:

The matched positions of sampling points are effected by other points. Point *c* is matched to "E Yesler Way" (the horizontal blue dot line) rather than "13th Ave" street though point *c* is much closer to "13th Ave" street. Why is that? This is because when we are labeling point *c*, we have considered, subconsciously, the positions of the other points nearby (e.g. point *a, b, d, e*) and their tendency. We can definitely decide that *c* is not located at "13th Ave" (the vertical road). In other words, we regard the neighboring points as point *c*'s reference points. The sampling points affect each other.

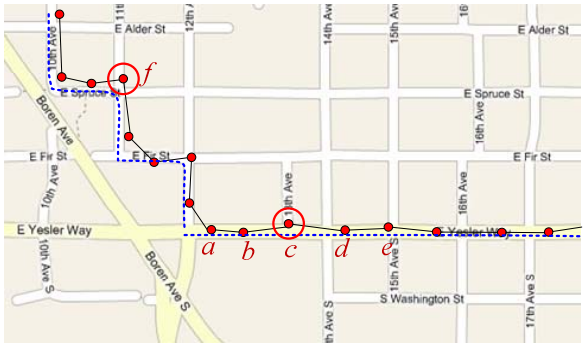


Figure 2. Illustation of our insights

2) Mutual Influence:

The sampling points have an interactive influence on each other. As stated above, when we are labeling point *c*, we look upon its neighboring points as reference points. However, in the same time, point *c* is also other points' reference point when labeling its neighbors. Therefore, this reference relationship is mutual and interactive.

3) Weighted Influence:

The farther away two sampling points are, the more limited they influence each other. In Fig. 2, when we are labeling point *c*, as stated above, we refer to the positions of other points. However, the influence made by points *d* and *b* is obviously more important than point *f* which is farther away from point *c*. Whether point *f* matched to the vertical road or the horizontal road hardly affects the position of point *c*. This fact indicates that the influence made by the reference points varies according to their distances from the matched point.

Based on these insights, we present a novel **Interactive Voting-based Map Matching** algorithm (termed IVMM). We design a **voting** process among all the sampling points to reflect their interactive influence. For each sampling point, we find out their candidate road segments, and for each candidate, there exists an optimal path which is passing through it. Every candidate will vote for their "best path", and the global optimal path will be chosen according to the voting result.

In general, our contributions can be summarized as:

- We study the interactive influence of the GPS tracking points and propose a novel voting-based algorithm IVMM for map matching low-sampling-rate GPS tracking data.
- Extensive experiments are conducted on real datasets. The data is collected from real-world and labeled by real people. Therefore, the results are more reliable than synthetic data used in most existing work
- The evaluation results validate the advancement of our IVMM algorithm compared to existing method for low-sampling rate data in terms of matching accuracy.

The outline of this paper is as follows. Section II reviews the related work and categories current map matching algorithms. Section III formulates the map matching algorithm and gives an overview of IVMM algorithm. The detail of IVMM algorithm is introduced in Section IV and analyzed in Section V. The experiment results are presented in Section VI. Section VII concludes the paper and gives directions for our future work.

II. RELATED WORK

This section reviews the existing map matching algorithms and categories them. Approaches for these algorithms can be classified by various criteria.

A. Involved Information of Input Data.

According to the information of input GPS tracking data used, existing methods can be categorized into four groups: *geometric* [12], *topological* [13, 14], *probabilistic* [15] and other *advanced* techniques [16, 17, 19]. Geometric map matching algorithms utilize the geometric information of the spatial road network data by considering only the shape of the links without the connectivity of the links [12]. A map matching algorithm which makes use of the the connectivity and contiguity information is referred to as a topological map matching algorithm [14]. Topological methods use the topology of map features to constrain the candidate matches for a sampling point. For example, in Fig. 3, though point *A* is closer to the vertical road, but *A* is sure to be matched to "Redmond Way" since there is no way to go from the vertical road to the "Redmond Way" rather than the vertical road. A probabilistic map matching algorithm is developed in [15]. Advanced map matching algorithms are referred to as those using more refined concepts such as a Kalmam Filter [16], a fuzzy logic model [17] or the application of Hidden Markov Model [19, 29]. However, these algorithms still perform poor when the sampling rate is low. In [29], the evaluation shows the error rate is more than 50% when the sampling interval exceeds 5 minutes.

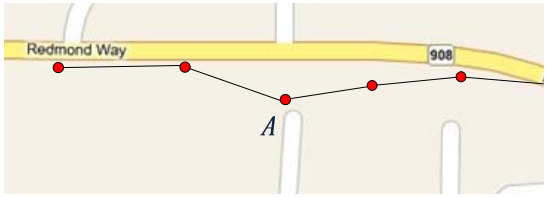


Figure 3. Map matching using topological information

Our approach is based on both geometric and topological information of the road network. The IVMM algorithm, different from existing topological methods, also considers the temporal/speed information of the road network.

B. Global Algorithms and Local Algorithms

Map matching algorithms can also be categorized into *local/incremental* algorithms and *global* algorithms according to the range of sampling points considered when matching the trajectories.

The local/incremental algorithms follow a greedy strategy of sequentially extending the solution from an already matched portion. These methods try to find a local optimal point or edge based on distance and orientation similarity [12, 18, 20]. The incremental algorithm described in [21] using a constant-depth recursive look-ahead strategy is also based on locally matching geometries. Wenk et al. proposed an "adaptive clipping" method which utilizes the Dijkstra algorithm to construct a shortest graph on local free space graph [22]. A segment-based matching algorithm introduced in [23] assigns confidence values for different sampling points. This algorithm matches high-confidence segments first, and then matches low-confidence segments using previous matched edges. In general, local/incremental algorithms usually run fast when sampling rate is very high (e.g. 2-5 seconds a sample). Therefore, the local/incremental algorithms are often adopted in the applications with *online* requirement. However, the performance of these algorithms gets worse when the sampling rate is not high. As the sampling rate decreases, the problem of "arc-skipping" [12] becomes prominent, causing significant degrade of accuracy.

The other group of algorithms called global algorithms tries to find a trajectory which is as closer as the sampling track among all available trajectories in the road network. To measure the similarity between the matched trajectory and the sampling trajectory, most algorithms employ the "Fréchet distance" or "Weak Fréchet distance". The algorithm proposed in [5] applies parametric search over all critical values, then it solves the decision problem by finding a monotone path in the free space from the lower left corner to the upper right corner. This work is extended in [21] with average Fréchet distance to reduce the effect of outliers. Paper [21] also uses weak Fréchet distance that runs in $O(mn \log mn)$ time with similar matching quality. Paper [14] proposes an algorithm based on a weighted graph representation of the road network. This algorithm is based on a measurement similar to the "Average Fréchet distance". Since the global algorithms require the information of an overall tracking trajectory, they are often used in offline situation.

In contrast to current global map matching algorithms, we consider the mutual influence of the sampling points themselves as well as the impact of remoteness on the positions of the sampling points. Based on a novel voting strategy, our algorithm is more robust and reliable than existing methods.

C. Sampling Rate

According to the sampling density of the tracking data, we can categorize existing map matching algorithms to dense-sampling-rate approaches and low-sampling-rate approaches. As stated in [14], most global algorithms perform poor when GPS sampling intervals are larger than 120 seconds- the average correctness method is less than 60%. To the best of our knowledge, [1] is the first and only paper to address the problem of map matching for low-sampling-rate GPS trajectories. **ST-Matching** [1] is a map matching algorithm for low-sampling-rate GPS tracking data which incorporates both the geometric topologic structure and the speed constrains. This algorithm first retrieve a set of candidate points for each sampling point, then define a similarity function with respect to every two sequential sampling points and their candidate points. Based on the summation of the similarity function during the whole path, the ST-Matching algorithm finds a path with the largest summation to be the result path. The authors conducted extensive evaluation to investigate the performance of ST-Matching algorithm. The experiment results validate ST-Matching outperforms the methods based on Fréchet distance both in terms of matching accuracy and running time.

However, by experiments, we find that when the length of trajectory is long and the vehicle passes through a multi-lane road, the matching result given by ST-Matching algorithm is still not reliable. For example, Fig. 5 is a screenshot of the visualized result of ST-Matching algorithm on a real GPS tracking data. The red curve connects the GPS sampling data and the blue line marked by green pushpins is the matching result of ST-Matching algorithm with respect to the sampling data. The matched path still runs in a roundabout zigzag way which obviously deviates from the real path.

Figure 4. A screenshot of ST-Matching result

The mismatching of ST-Matching algorithm is caused by these reasons:

- The similarity function is built solely with respect to two adjacent candidate points, whereas the position of a sampling point is influenced by all its neighboring points, both previous points and latter points. Thus the values of the edges in the candidate graph computed by ST-Matching algorithm do not reflect the true influence of the neighboring points enough.
- ST-Matching algorithm uses a simple summation of all the values in the trajectory to evaluate the similarity of a candidate path with the sampling data. It neglects the impact of remoteness on the sampling point by other points.
- The reciprocal influence mentioned in Section I (our insight 3) is not considered by ST-Matching algorithm.

If one point is matched to a wrong road segment, the following points will all be measured based on this mismatched point, thus that the errors of the process are cumulative.

III. PRELIMINARY

In this section, we first define the concepts used in this paper and then give an overview of the interactive voting-based map matching algorithm.

A. Problem Definition

Definition 1. GPS trajectory: A GPS trajectory T is a point sequence $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$ linked by the time stamps of the GPS points. Each GPS point p_i is a triple $(p_i.\mathrm{lat}, p_i.\mathrm{lng}, pi.\mathrm{t})$ which are its latitude, longitude and timestamp respectively.

Definition 2. Road network: A road network is a directed graph $G(V, E)$, where E is a set of edges representing the road segments. V is a set of vertices representing the intersections and terminal points of the road segments. Each road segment e is a directed edge that is associated with an id $e.\mathrm{eid}$, a typical travel speed $e.\mathrm{v}$, a length value $e.\mathrm{l}$, a starting point $e.\mathrm{start}$, an ending point $e.\mathrm{end}$ as shown in Fig. 5.

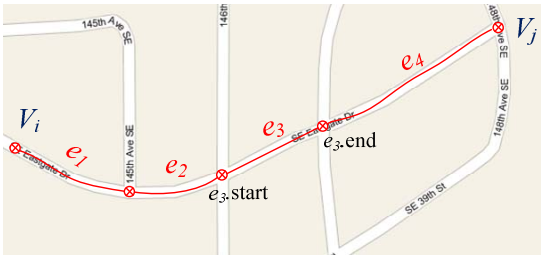


Figure 5. Illustration of a path in a road network

Definition 3. Path: Given two vertices V_i, V_j in a road network G, a path P is a set of connected road segments that start at V_i and end at V_j, i.e. $P: e_1 \rightarrow e_2 \rightarrow \cdots \rightarrow e_n$, where $e_1.start = V_i$, $e_n.end = V_j$, $e_k.end = e_{k+1}.start$, $1 \leq k < n$. Fig. 5 illustrate a typical path connecting several road segments.

The map matching problem then can be defined as:

Given the road network G and a raw GPS trajectory T, find a path in G which matches T with its real path.

B. Framwork

Motivated by the insights we presented in section I as well as the disadvantages of ST-Matching we discussed in section II, we develop the IVMM algorithm which is aimed to make the best of the interactive relationship among all the sampling points so as to find a global optimal path to match the trajectory, especially for low-sampling-rate situation. The IVMM algorithm consists of four phases: *candidate preparation*, *score matrix building*, *interactive voting* and *path finding*. Fig. 6 shows a framework of the IVMM algorithm.

In the first phase, we perform a range query to select the candidate road segments (CRS) and candidate points (CP) for each sampling point, then in the second phase, we construct a candidate graph by performing a spatial and temporal analysis of the position context. After that, we build a static score matrix according to the candidate graph. The mutual influence is modeled utilizing a weighted score matrix which is constructed dynamically. Based on the weighted score matrix, all the candidate points parallelly vote for their best matching paths in the last phase. Then a global optimal path is elected according to the voting result.

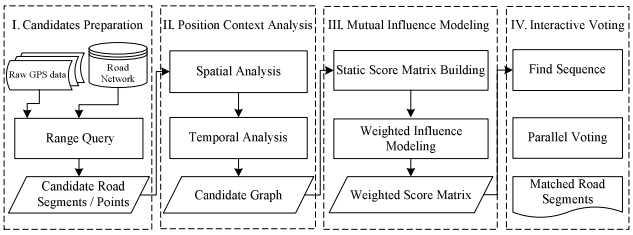


Figure 6. Overview of the IVMM algorithm

IV. INTERACTIVE VOTING-BASED MAP MATCHING ALGORITHM

In this section, we introduce our Interactive Voting-based Map Matching Algorithm (IVMM) and give an example to show how IVMM works.

A. Candidates Preparation

Given a GPS trajectory $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, we retrieve a set of **candidate road segments (CRS)** for each sampling point by a *range query*. The set CRS_i contains all road segments within radius r (a fixed number) of p_i with respect to Euclidian distance.

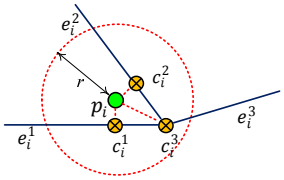


Figure 7. Illustration of candidate road segements and candidate points

The **candidate points (CP)** are selected in the following way: if the projection of the sampling point onto the road segment is between its endpoints, then choose the geometry projection as the candidate point; otherwise, choose the endpoint which is closer to the sampling point with regard to the Euclid distance as shown in Fig. 7. Thus we get a candidate point set $\mathrm{CP_i}$ for each p_i. Fig. 8 gives an example of a trajectory $p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_4$. After the process of candidates preparation, we obtain **CRS** and **CP** for each p_i depicted in Fig. 8. For instance, $\mathrm{CRS_1}=\{e_1^1, e_1^2, e_1^3\}$ and $\mathrm{CP_1}=\{c_1^1, c_1^2, c_1^3\}$. Let a_i be the **cardinality** of CRS_i, thus we have $a_i = |\mathrm{CRS}_i| \leq |\mathrm{CP}_i|$.

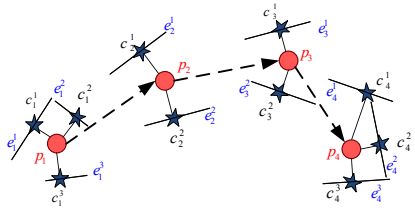


Figure 8. Candidate road segments and candidate points

B. Position Context Analysis

In this phase, we perform the *Spatial Temporal Analysis* and construct a *candidate graph* $G'(V', E')$. The distribution of the measurement error is assumed to satisfy the Gaussian distribution $N(\mu, \sigma^2)$ [28]. For a candidate point c_i^j, its observation probability with respect to p_i is formulated as:

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

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

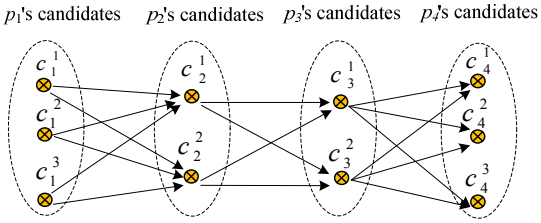


Figure 9. Candidate graph

From candidate point c_{i-1}^t to candidate point c_i^s, the *spatial analysis function* is defined as below:

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

where $V(c_{i-1}^t \to c_i^s)$ is the transition probability 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 $i-1$ to sampling point 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. V' is the union of $\mathbf{CP}_i$ and E' is a set of shortest paths between any two candidate points. The spatial analysis function is aimed to measure the similarity between the candidate paths with the shortest path with respect to two adjacent candidate points. This is based on the assumption that a driver is more likely to choose a shorter route when driving. We also define a *temporal analysis function* considering the speed constrains of the road segments as follows:

$$F_t(c_{i-1}^t \to c_i^s) = \frac{\sum_{u=1}^{k}(e'_u.v \times \bar{v}_{(i-1,t)\to(i,s)})}{\sqrt{\sum_{u=1}^{k}(e'_u.v)^2} \times \sqrt{\sum_{u=1}^{k} \bar{v}^2_{(i-1,t)\to(i,s)}}} \quad (4)$$

where $e'_1 \to e'_2 \cdots \to e'_k$ is the shortest path connecting c_{i-1}^t with c_i^s, and $\bar{v}_{(i-1,t)\to(i,s)}$ the average speed from c_{i-1}^t to c_i^s which can be easily computed. We adopt the definition of ***ST-function*** for $c_{i-1}^t \to c_i^s$ in [1] which is given as below:

$$F(c_{i-1}^t \to c_i^s) = F_s(c_{i-1}^t \to c_i^s) \times F_t(c_{i-1}^t \to c_i^s) \quad (5)$$

where F_s is the *spatial analysis function* and F_t is the *temporal analysis function*. (For the details of spatial/ temporal analysis, please refer to paper [1].) After spatial and temporal analysis, a candidate graph is constructed. The nodes of the graph are the set of candidate points for each GPS observation, and the edges of the graph are set of shortest paths between any two neighboring candidate points. The nodes and edges are all assigned weight values based on the results of spatial/temporal analysis.

C. Mutual Influence Modeling

1) Static Score Matrix Building

Based on the candidate graph generated in the previous phase, we can build a **Static Score Matrix** denoted as $\boldsymbol{M} = diag\{\boldsymbol{M^{(2)}}, \boldsymbol{M^{(3)}}, \dots, \boldsymbol{M^{(n)}}\}$ where $\boldsymbol{M^{(i)}} = (m_{ts}^{(i)})_{a_{i-1} \times a_i} = (F(c_{i-1}^t \to c_i^s))_{a_{i-1} \times a_i}$. We provide the score matrix of trajectory $p_1 \to p_2 \to p_3 \to p_4$ as an example.

$$\boldsymbol{M} = \begin{bmatrix} 0.8 & 0.6 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.7 & 0.5 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.6 & 0.4 & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.3 & 0.7 & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.2 & 0.4 & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & 0.3 & 0.5 & 0.4 \\ -\infty & -\infty & -\infty & -\infty & 0.6 & 0.7 & 0.9 \end{bmatrix}$$

Each item in this score matrix represents the probability of a candidate point like c_2^1 to be a correct projection only considering the information of two consecutive points, e.g., $c_1^1 \to c_2^1$. But, this information does not reflect the interactive mutual influence between the candidate points, as discussed in Section II.

2) Weighted Influence Modeling

To model the weighted influence of the candidate points, we define a $(n-1)$-dimension *Distance Weight Matrix* $\boldsymbol{W_i}$ for each sampling point p_i. This diagonal matrix gives weights for the effect of all other points to p_i associated with their distances to p_i. For $= 2,3, \dots n$, $\boldsymbol{W_i}$ defined as below:

$$\boldsymbol{W_i} = diag\left\{w_i^{(1)}, w_i^{(2)}, \dots, w_i^{(i-1)}, w_i^{(i+1)} \dots, w_i^{(n)}\right\} \quad (6)$$

and $\boldsymbol{W_1} = \left\{w_i^{(2)}, w_i^{(3)}, \dots, w_i^{(n)}\right\}$ (7)

where $w_i^{(j)} = f(dist(p_i, p_j)) \quad j = 1,2, \dots, n$

and $dist(p_i, p_j)$ is the Euclidian distance between p_i and p_j. We term the f function as *distance weight function*, which represents the influence of the point p_j to the point p_i based on the distance between them. We'll further discuss the f function later in Section V.

Regarding matrix $\boldsymbol{M}$ as a $(n-1)$-order block matrix, for each $i = 1,2,3, \dots, n$, the **Weighted Score Matrix** is defined as:

$$\boldsymbol{\Phi_i} = \boldsymbol{W_i M} = diag\left\{\boldsymbol{\Phi_i^{(2)}}, \boldsymbol{\Phi_i^{(3)}}, \dots, \boldsymbol{\Phi_i^{(n)}}\right\} \quad (8)$$

For each candidate point of sampling point p_i, matrix $\boldsymbol{\Phi_i}$ represents the similarity of all the candidate road segments with the true path considering the influence of the remoteness. Note this matrix multiplication process ensures that if two sampling points have the same distances to the third sampling point, their

influences on the third sampling point are also equal. That is because matrix $\boldsymbol{W_i}$ is an $n-1$ order matrix just without the element $w_i^{(i)}$. In fact, for $j = 2,3,\dots,n$, $\boldsymbol{\Phi_i^{(j)}} = \left(\varphi_{ts}^{(i,j)}\right)_{a_{j-1}\times a_j}$

$$= \begin{cases} w_i^{(j-1)}\boldsymbol{M}^{(j)} = (w_i^{(j-1)} m_{ts}^{(j)})_{a_{j-1}\times a_j} & \text{if } 1 \le j \le i \\ w_i^{(j)}\boldsymbol{M}^{(j)} = (w_i^{(j)} m_{ts}^{(j)})_{a_{j-1}\times a_j} & \text{otherwise} \end{cases} \quad (9)$$

To illustrate this processing, let's follow with the previous example. If we set $f = 2^{-(dist(p_i,p_j))}$, and suppose $dist(p_i,p_j) = |i-j|$ (in real implementation, the *dist* function is the actual Euclid distance, here just for convenience of explanation), the weighted score matrix $\boldsymbol{\Phi_i}$ are:

$$\boldsymbol{\Phi_1} = \begin{bmatrix} 0.4 & 0.3 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.35 & 0.25 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.3 & 0.2 & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.075 & 0.175 & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.05 & 0.1 & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & 0.038 & 0.063 & 0.05 \\ -\infty & -\infty & -\infty & -\infty & 0.075 & 0.088 & 0.113 \end{bmatrix}$$

$$\boldsymbol{\Phi_2} = \begin{bmatrix} 0.4 & 0.3 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.35 & 0.25 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.3 & 0.2 & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.15 & 0.35 & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.1 & 0.2 & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & 0.075 & 0.125 & 0.1 \\ -\infty & -\infty & -\infty & -\infty & 0.15 & 0.175 & 0.225 \end{bmatrix}$$

$$\boldsymbol{\Phi_3} = \begin{bmatrix} 0.2 & 0.15 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.175 & 0.125 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.15 & 0.1 & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.15 & 0.35 & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.1 & 0.2 & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & 0.15 & 0.25 & 0.2 \\ -\infty & -\infty & -\infty & -\infty & 0.3 & 0.35 & 0.45 \end{bmatrix}$$

$$\boldsymbol{\Phi_4} = \begin{bmatrix} 0.1 & 0.075 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.088 & 0.063 & -\infty & -\infty & -\infty & -\infty & -\infty \\ 0.075 & 0.05 & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.075 & 0.175 & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0.05 & 0.1 & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & 0.15 & 0.25 & 0.2 \\ -\infty & -\infty & -\infty & -\infty & 0.3 & 0.35 & 0.45 \end{bmatrix}$$

For instance, matrix $\boldsymbol{\Phi_2}$ is the weighted score matrix for sampling point p_2. Compared with the static matrix $\boldsymbol{M}$, $\boldsymbol{\Phi_2^{(2)}}$ and $\boldsymbol{\Phi_2^{(3)}}$ are both halved since $\boldsymbol{\Phi_2^{(2)}}$ is associated with the transmission from CP_1 to CP_2, and $\boldsymbol{\Phi_2^{(3)}}$ is related to the transmission from CP_2 to CP_3. Based on the assumed distance weight function $|i-j|$, their impacts on p_2 are the same.

The static score matrix is constructed in this phase; whereas in practice, the distance weight matrix and weighted score matrix can be generated dynamically in the next phase when needed so as to save the memory space though we defined them here.

D. Interactive Voting

Recall the insights presented in Section I, the influence of the sampling points is mutual. When a sampling point refers to other points, it is referenced by other points in the same time. Considering the interaction of the candidate points, we design a novel and efficient voting algorithm for the candidate selection. For each sampling point p_i, the weighted score matrix is built. Then for each candidate point c_i^k, we tried to find an optimal path P which passes across c_i^k utilizing the weighted score matrix $\boldsymbol{\Phi_i}$. After that, candidate point c_i^k votes for all the candidate points on path P. Since the entire voting process is independent for each sampling point, it can be implemented in parallel. We now present our algorithm in pseudo code.

Algorithm 1 InteractiveVoting

Input: Candidate graph $G'(V', E')$, static score matrix $\boldsymbol{M}$
Output: The vote counts of each candidate point
1: **for** i = 1 **to** n **do in parallel**
2: Compute $\boldsymbol{W_i}$ and $\boldsymbol{\Phi_i}$
3: **for** k = 1 **to** a_i **do**
4: P=FindSequence (G', $\boldsymbol{\Phi_i}$, i, k)
5: Vote for each candidate point in trajectory P

For each candidate of p_i, the FindSequence process is called to find a path P which passes through the candidate point and has the largest weighted score summation. The FindSequence process works in a dynamic-programming paradigm. The pseudo-code is as follows:

Algorithm 2 FindSequence

Input: Candidate graph $G'(V', E')$, matrix $\boldsymbol{\Phi_i}$, i, k
Output: The sequence P: $c_1^{s_1} \to \cdots c_i^k \to \cdots \to c_n^{s_n}$
/* retrieve a path which must get across c_i^k and has the largest weighted score summation with respect to $\boldsymbol{\Phi_i}$ */
1: Let $f_{i,k}[\,]$ denote the highest score computed so far;
2: Let $pre_{i,k}[\,]$ denote the parent of current candidate;
3: **for** t = 1 **to** a_1 **do**
4: $f_{i,k}[c_1^t] = w_i^{(1)} N(c_1^t)$;
5: **for** s = 1 **to** a_i **do**
6: **if** $s \neq k$ **then**
7: **for** t = 1 **to** a_{i-1} **do**
8: $\varphi_{ts}^{(i,i)} = -\infty$
5: **for** $j = 2$ **to** n **do**
6: **for** s = 1 **to** a_j **do**
7: $f_{i,k}[c_j^s] = \max\{ f_{i,k}[c_{j-1}^t] + \varphi_{ts}^{(i,j)}, t = 1,2,\dots a_{j-1}\}$
8: $pre_{i,k}[c_j^s] = \text{argmax}_{c_{j-1}^t}\{ f_{i,k}[c_{j-1}^t] + \varphi_{ts}^{(i,j)}, t = 1,2,\dots a_{j-1}\}$
9: Initialize $rList$ as an empty list; /* a list of matched points in reverse order*/
10: $c = arg\,max_{c_n^s}(f_{i,k}[c_n^s], s = 1,2,\dots a_n)$;
11: $c_i^k.fValue = max(f_{i,k}[c_n^s], s = 1,2,\dots a_n$
12: **for** $i = n$ to 2 **do**
13: $rList$.add(c);
14: $c = pre_{i,k}[c]$;
15: $rList$.add(c);
16: **return** $rList.reverse()$;

The FindSequence process retrieves the best path for each candidate point (we call it a *local optimal path*) as well as a *fValue* which is used in the next phase when two candidates have the same number of votes.

Now continue with our example before. As stated in the second phase, each candidate point has an observation probability. Given the observation probability of $\mathbf{CP}_1$ as shown in Tab. 1, after the Find Sequence process, the best path for candidate c_3^1 and its *fValue* are illustrated in Fig. 10.

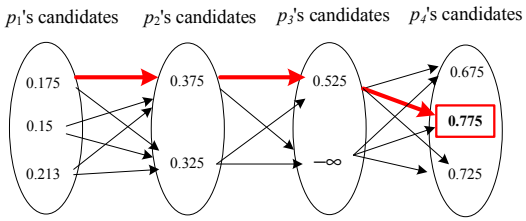


Figure 10. FindSequence process of candidate c_3^1

TABLE I. INITIAL OBSERVATION PROBABILITY

CP	c_1^1	c_1^2	c_1^3
$N(c_1^i)$	0.7	0.6	0.85

After the interactive voting, we pick the candidate point who has the most votes for each sampling point p_i; if two candidate point have the same votes, ties are broken using the *fValue*. Then we link the candidate points one by one, the global optimal path is obtained.

TABLE II. VOTING RESULT

CP	c_1^1	c_1^2	c_1^3	c_2^1	c_2^2	c_2^3	c_3^1	c_3^2	c_4^1	c_4^2	c_4^3
Votes	8	1	1	9	1	1	9	1	2	7	
fValue	1.4	1.2	1.4	1.3	1.1	0.8	1.2	0.7	0.7	0.8	

Considering the above example, after Interactive Voting, we count the votes for each candidate point. The result shows in Tab. II. Then the global optimal path is elected which is $c_1^1 \rightarrow c_2^1 \rightarrow c_3^2 \rightarrow c_4^3$ as lined out in Fig. 11.

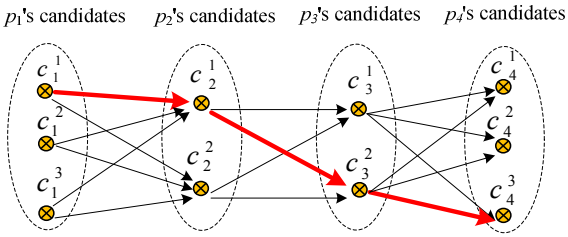


Figure 11. Matching result of IVMM algorithm

V. ALGORITHM ANALYSIS

This section first analyzes the complexity of IVMM algorithm, and then discusses the selection of the distance weight function.

A. Algorithm Complexity

The time complexity of IVMM can be divided into four parts according to the four phases of IVMM. Let n denotes the number of sampling points on the given trajectory, and m denotes the number of road segments in the road network. We further assume that the maximum number of candidates of one sampling point is k.

In the first phase, a range query is performed to get the candidate points. The time cost for range query is $O(4nk)$ by using cell index [24]. In the second phase, the construction of the candidate graph takes a $O(nk^2 m \log m)$ time since the number of shortest paths in G_T' is $(n-1)k^2$ and the shortest path can be computed in $O(m \log m)$ time utilizing Dijkstra algorithm (n ≪ m). The weighted score matrix can be built in $O(nk^2)$ time. The complexity of FindSequence procedure is $O(nk^2)$ since each candidate is at most visited once. Therefore, the Interactive Voting algorithm takes a $O(nk^2)$ time due to parallel computation. Finally, the complexity of the last phase is $O(nk^2)$. In total, the complexity of IVMM is $O(nk^2 m \log m)$. Note that even if the Interactive Voting algorithm uses a round-robin way rather than a parallel computation, the total complexity of IVMM is still $O(nk^2 m \log m)$ because usually in a single trajectory n ≪ m for low-sampling-rate scenario. Note that the complexity of ST-Matching algorithm and IVMM algorithm is the same. [1]

B. The Distance Weight Function

In Section IV, we introduce a *distance weight function* (f) in the position context analysis phase of IVMM algorithm. Let x be the Euclidian distance of two sampling points, then $f_r(x)$ denotes the impact of distance x on each sampling point with respect to road network r. The chosen of f is an interesting open problem. However, the f function should satisfy the following properties:

1) $f_r(0) = 1$.
2) $f_r(\infty) = 0$.
3) For $\forall x > 0,\ 0 < f_r(x) < 1$

In general, f should be a declining function since the farther away, the less two sampling points impact on each other. For example, an exponential declining function e^{-x} is an available distance weight function.

We believe that the impact of the distance on the position of the sampling points satisfies Gassian/Normal distribution since if two sampling points are relatively close, then their influence on another point is pretty much the same, while if the distance of two sampling points exceeds a certain limit, the influence of each sampling point declines very fast. Therefore, in our work, the distance weight function is defined as:

$$f(x) = e^{-\frac{x^2}{\beta^2}} \tag{10}$$

where β is a parameter with respect to the road network. In Section VI, we investigate the impact of parameter β and the impact of different distance weight functions.

VI. EVALUATION

In this section, we conduct extensive experiments to evaluate our IVMM algorithm. We first describe the settings, then introduce the evaluation approach, and lastly report the evaluation results.

A. Setting

1) Road Netwroks

In our experiments, we use the road network of Beijing visualized in Fig. 12. The network graph contains 58,624 vertices and 130,714 road segments. The vertical length of the map is about 47.7 km and horizontal length is about 52.6 km.

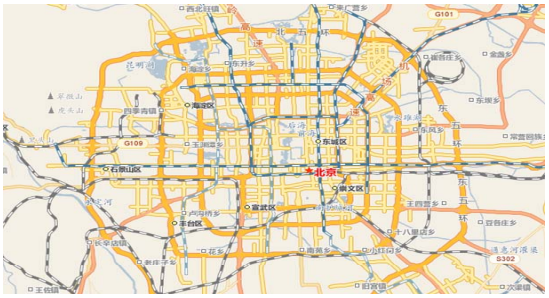
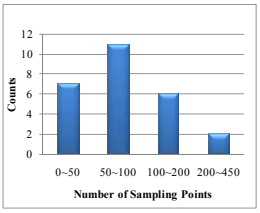


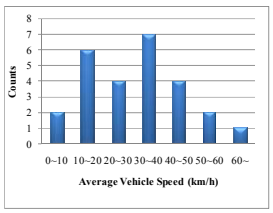
Figure 12. Road network of Beijing

2) GPS Tracking Data

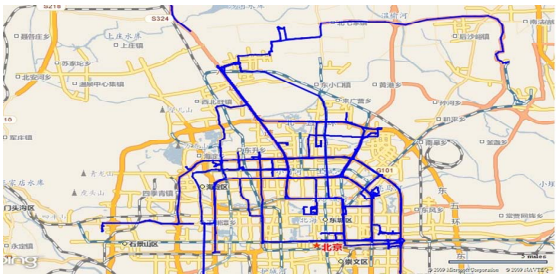
In our experiments, all GPS tracking data is real data collected from the Geolife [25, 26] System. Distinctive from existing approaches, we use real human labeled *true path* data as the ground truth in our experiments. The raw GPS data set contains 26 trajectories with varying number of points and average speed as Fig. 13(a) and Fig. 13(b) present. The trajectories cover about 643 kilometers over more than 30 hours. Fig. 13(c) plots all the original trajectories on the map.



a) Statistics on number of points



b) Statistics on average vehicle speed



c) Screenshot of all the GPS trajectories in the dataset

Figure 13. Distribution of the GPS trajectories

3) Parameter Selection

The sampling rate: This work focus on the low-sampling-rate scenario, thus the sampling interval ranges from 30 seconds to 10.5 minutes in our experiments.

Parameters for IVMM algorithm: In our experiments, we set k=5 as the maximum number of candidates for each sampling point. The radius of the range query is set to 100 meters. For observation probability estimation, we use a normal distribution (Equation (1)) with $\mu = 5$ meters and $\delta = 10$ meters.

The distance weight function: We use the distance weight function defined as equation (10) for the estimation. The parameter β is set to 7 km as the default set. We also investigate the impact of different choice of parameter β and some other distance weight functions on the matching result.

Platform: The algorithms are implemented in Java, on an Intel 2.33 GHZ PC with 2GB memory on Windows 7 operating system.

B. Evaluation Approach

Since the ST-Matching algorithm is the only map matching algorithm which performs well for low-sampling-rate data [1], we compared our IVMM algorithm with ST-Matching both in terms of matching quality and efficiency. To evaluate the efficiency of our algorithm, we compared the running time of both algorithms. To evaluate the matching quality, we use the *correct matching percentage* (CMP) calculated by the following equation:

$$\text{CMP} = \frac{\text{Correct matched points}}{\text{Number of points to be matched}} \times 100\% . \qquad (11)$$

We also investigate the impact of different distance weight functions on the matching quality.

C. Results

1) Vitualized results

Fig. 14 and Fig. 15 present the screenshots of the visualized matching results of IVMM algorithm compared with ST-Matching algorithm on the same GPS tracking data. The red line in Fig. 14 represents the GPS tracking data and the blue line marked with red pushpins shows the matching results. The left picture of Fig. 14 is the result of ST-Matching algorithm. It runs in a roundabout way which obviously deviates from the true path. The right picture is the matching result of IVMM algorithm. It matches all ground truth road segments in this portion.

Figure 14. A screenshot of maching result (ST-Maching V.S. IVMM)

Fig. 15 is another screenshot of the visualized matching result. The original GPS tracking data is the same with the example presented in Fig. 4. In both of the two pictures below, the red line represents the original tracking data and blue line marked with green pushpins is the matching result. On the left, ST-Matching mismatches several road segments such that the result is a "zigzag" way whereas on the right our IVMM "straightens" the mismatched road segments and matches the ground truth path quite well. Compared with ST-Matching algorithm, the result of our IVMM algorithm is more reliable and adaptive.

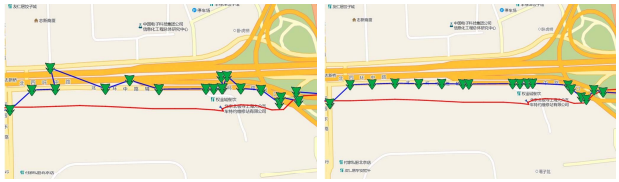


Figure 15. Another screenshot of matching result (ST-Matching V.S. IVMM)

2) Correct Matching Percentage

Fig. 16 presents the matching accuracy comparison result based on the correct matching percentage (CMP). It is clearly that IVMM algorithm significantly outperforms ST-Matching algorithm for all sampling intervals. Recall the statistic result of real taxi GPS data in Beijing (Fig. 1), the sampling interval of most low-sampling-rate GSP trajectories are 2~6 minutes (about 86% among all low-sampling-rate data). When sampling interval ranges from 1.5 minutes to 6.5 minutes, the accuracy of IVMM is always about 70% with a 10% improvement of ST-Matching algorithm. When the sampling interval exceed 6.5 minutes, the performances of IVMM and ST-Matching get closer but still have a 5% margin. The result validates that IVMM algorithm represents the interactive influence of each sampling points effectively and is more robust compared with ST-Matching algorithm.

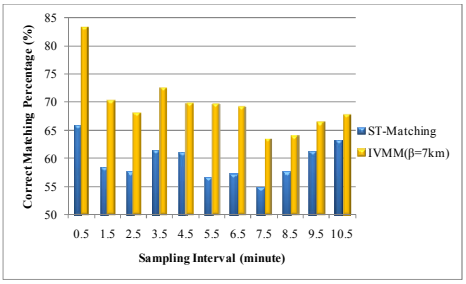


Figure 16. Matching accuracy (ST-Matching V.S. IVMM)

3) Running Time

The time complexity of IVMM algorithm is the same with ST-Matching as analyzed in Section V. We conducted experiments to evaluate the running time of both algorithms. The bar chart in Fig. 17 gives the result.

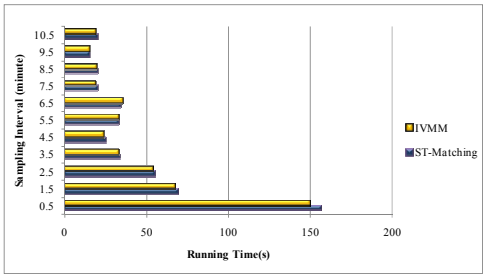


Figure 17. Running time (ST-Matching V.S. IVMM)

In this evaluation, the sampling interval ranges from 0.5 minutes to 10.5 minutes. The result demonstrates that IVMM algorithm is as fast as ST-Matching algorithm for both low-sampling-rate and high-sampling-rate data. Fig. 17 also implies that as the sampling interval increases, the number of points to be matched is correspondingly reduced such that time cost of both algorithms decreases fast.

4) Impact of Distance Weight Functions

In Section V, we discuss the choice of distance weight functions. Fig. 18 plots some different distance functions we evaluate on the same data set. The x-axis is the distance of two sampling points; y-axis is the value of distance weight function which represents the influence on the positions of two sampling points.

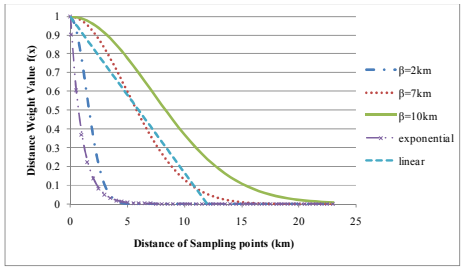


Figure 18. Different distance weight functions

Fig. 19 presents the matching performance of different weight functions. The exponential function and linear function perform worse than the Gaussian function in terms of matching accuracy. The performance of IVMM algorithm without distance weight function (denoted as IVMM (none) in Fig. 19) is also worse than approaches with a Gaussian distance weight function for all sampling intervals. This validates again that the influence of sampling points is related to their distance.

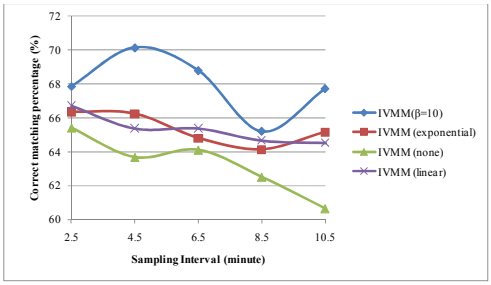


Figure 19. Impact of different distance weight functions

We also estimate the impact of β on the distance weight function as shown in Fig. 20. The accuracy rate when $\beta = 2\ km$ is lower than when $= 4km, 7km$ and $10km$, but still better than ST-Matching algorithm in Fig. 16. The reason for this is that when $\beta = 2km$, the value of distance weight function decreases too fast as showed in Fig. 18.

VII. CONCLUSION

In this paper, we investigate the problem of map matching for low-sampling-rate GPS trajectories. A novel algorithm termed IVMM is proposed and analyzed. This algorithm employs a voting-based approach to reflect the mutual influence of the sampling points. We define a distance weight function to evaluate the impact of distance to the matching positions. Extensive experiments are conducted with real GPS tracking data. Both for low-sampling-rate data and high-sampling-rate data, the correct matching percentage of IVMM algorithm is

higher than ST-Matching algorithm. In particular, when the sampling interval ranges from 2 to 6 minutes, the accuracy rate of IVMM algorithm always has a more than 10% improvement over the ST-Matching algorithm. The results demonstrate that IVMM algorithm significantly outperforms ST-Matching algorithm which is so far the only approach aimed at low-sampling-rate GPS data in terms of matching quality.

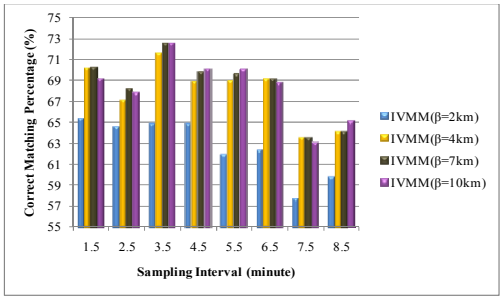


Figure 20. Impact of parameter β with IVMM

The distance weight function which plays an important role in IVMM algorithm is intriguing. We note that the weighted influence of the sampling points is not only related with the distance but also with the topology of the road networks. Typically, the influence of two sampling points in a dense road network is much less than that of a sparse one. That is because the more the number of possible paths traversing two points is, the less weighted influence they might brought on each other. We will do more research on this topic in our future work.

ACKNOWLEDGEMEN

This work is partially supported by Microsoft Research Asia Internet Services Theme Research Program.

REFERENCES

[1] C.Y. Zhang, Y. Zheng and X. Xie, "Map-Matching for Low-Sampling-Rate GPS trajectories", in Proceedings of ACM SIGSPATIAL Conference on Geographical Information Systems (ACM GIS 2009) .

[2] I. Rauschert, P. Agrawal, R. Sharma, S. Fuhrmann, I. Brewer and A. MacEachren, "Designing a Human-centered, Multimodal GIS Interface to Support Emergency Management", in Proceedings of the 10th ACM international symposium on Advances in geographic information systems, McLean, Virginia, USA, 2002.

[3] Bing Map Search. http://cn.bing.com/ditu. Microsoft Corporation

[4] D. Pfoser and C. S. Jensen, "Capturing the Uncertainty of Moving-Object Representations", In 6th International Symposium on Advances in Spatial Databases (Hong Kong, China, July 1999), SSD 2009.

[5] H. Alt, A. Efrat, G. Rote, and C. Wenk, "Matching Planar Maps", J. of Algorithms, vol. 49: pp.262–283, 2003.

[6] A. Civilis, C. S. Jensen, and S. Pakalnis, "Techniques for Efficient Road-network-based Tracking of Moving Objects", IEEE Transactions on Knowledge and Data Engineering, vol. 17(5): pp.698–711, 2005.

[7] M. A. Quddus,W. Y. Ochieng and R. B. Noland, "Current Map-Matching Algorithms for Transport Applications: State-of-the Art and Future Research Directions", Transportation Research Part C: Emerging Technologies 15 (2007), pp. 312–328

[8] R. R. Joshi, "A New Approach to Map Matching for In-vehicle Navigation Systems: the Rotational Variation Metric", In proceedings of IEEE Intelligent Transportation Systems.

[9] C. Feijoo, J. Ramos, and F. Perez,, "A System for Fleet Management Using Differential GPS and VHF Data Transmission Mobile Networks", Veh.0icle Navigation and Information Systems Conference, 1993., in Proc of the IEEE-IEE

[10] D. Bernstein and A. Kornhauser, "An Introduction to Map Matching for Personal Navigation Assistants", Technical report, NewJersey TIDE Center Technical Report,1996.

[11] M. A. Quddus, "Highintegrity Map-matching Algorithms for Advanced Transport Telematics Applications" ,PhDThesis. Centre for Transport Studies, Imperial College London,UK,2006.

[12] J. S. Greenfeld, "Matching GPS Observations to Locations on a Digital Map", In proceedings of the 81st Annual Meeting of the Transportaion Research Board, Wasington D. C, 2002.

[13] W. Chen, M. Yu, Z. Li and Y. Chen, "Integrated Vehicle Navigation System for Urban Applications", In proceedings of the 7th International Conference on Global Navigation Satellite System, pp. 15-22, 2003.

[14] H. B. Yin and O. Wolfson, "A Weight-based Map Matching Method in Moving Objects Databases1", In proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM'04), vol. 16, pp. 437-410, 2004.

[15] W. Y. Ochieng, M. A. Quddus and R. B. Noland, "Map-matching in Complex Urban Road Networks", Brazilian Journal of Cartography vol. 55(2), pp. 1-18, 2004

[16] D. Obradovic, H. Lenz and M. Schupfner, "Fusion of Map and Sensor Data in a Morder Car Navigation System", Journal of VSLI Signal Processing 45, pp. 112-122, 2006.

[17] M. A. Quddus, W. Y. Ochieng and R. B. Noland, "A High Accuracy Fuzzy Logic-based Map-Matching Algorithm for Road Transport", Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, vol. 10(3), pp. 103-115, 2006.

[18] A. Civilis, C. S. Jensen, J. Nenortaite and S. Pakalnis, "Efficient Tracking of Moving Objects with Precision Guarantees", in proc MobiQuitous conf., pp. 164-173, 2004.

[19] O. Pink and B. Hummel, "A Statistical Approach to Map Matching Using Road Network Geometry, Topology and Vehicular Motion Constraints", in proceedings of the 11th International IEEE Conference on Intelligent Transprotation Systems, 2008.

[20] A. Civilis, C. S. Jensen, J. Nenortaite, and S. Pakalnis, "Techniques for Efficient Road-network-based Tracking of Moving Objects", IEEE Transactions on Knowledge and Date Engineering, vol. 17(5), pp. 698-711, 2005.

[21] S. Brakatsouls, D. Pfoser, R. Salas and C. Wenk, "On Map-Matching Vehicle Tracking Data", In 31st International Conference on Very Large Data Bases (Trondheim, Norway,2005). VLDB 05.

[22] C. Wenk, R. Salas and D. Pfoser, "Addressing the Need for Map-Matching Speed: Localizing Globalb Curve-Matching Algorithms", In SSDBM 06: Proceedings of the 18th international Conference on Scientific and Statistical Database Management, 2006

[23] S.S. Chawathe, "Segment-based Map Matching", In IEEE Symposium on Intelligent Vehicles, 2007.

[24] J. L. Bentley and J. H. Friedman, "Data Structure for Range Searching", ACM Computing Surveys (CSUR), pp. 397-409, 1979.

[25] Y. Zheng, Y. Chen, X. Xie and W. Y. Ma, "GeoLife2.0: A Location-Based Social Networking Service", In proceedings of International Conference on Mobile Data Management 2009 (MDM 2009).

[26] Y. Zheng, L. Liu, L. Wan and X. Xie, "Learning Transportation Mode from Raw GPS Data for Geographic Application on the Web", In Proceedings of International conference on Wold Wild Web (WWW 2008), Beijing, China.

[27] Y. Zheng, L. Zhang, X. Xie and W. Y. Ma, "Mining Interesting Locations and Travel Sequences from GPS Trajectories", In Proceedings of International conference on Wold Wild Web (WWW 2009), Madrid Spain.

[28] F. van Diggelen, "GPS Accuracy: Lies, Damn Lies, and Staticstics", GPS World, vol. 9(1), pp. 41-45, 1998.

[29] P. Newson and John Krumm "Hidden Markov Map Matching Through Noise and Sparseness", in Proceedings of ACM SIGSPATIAL Conference on Geographical Information Systems (ACM GIS 2009).