2017 IEEE 7th International Symposium on Cloud and Service Computing

# A Hidden Markov Model-based Map-Matching Approach for Low-Sampling-Rate GPS Trajectories

Yu-Ling Hsueh, Ho-Chian Chen, and Wei-Jie Huang
Dept. of Computer Science & Information Engineering and Advanced Institute of Manufacturing with High-tech Innovations, National Chung Cheng University, Taiwan
{hsueh@cs.ccu.edu.tw, rainywindsnow@yahoo.com.tw, hwc105m@cs.ccu.edu.tw}

***Abstract*—Map matching is the process of matching a series of recorded geographic coordinates (e.g., a GPS trajectory) to a road network. Due to GPS positioning errors and the sampling constraints, the GPS data collected by the GPS devices are not precise, and the location of a user cannot always be correctly shown on the map. Unfortunately, most current map-matching algorithms only consider the distance between the GPS points and the road segments, the topology of the road network, and the speed constraint of the road segment to determine the matching results. In this paper, we propose a spatio-temporal based matching algorithm (STD-matching) for low-sampling-rate GPS trajectories. STD-matching considers the spatial features such as the distance information and topology of the road network, the speed constraints of the road network, and the real-time moving direction which shows the movement of the user. In our experiments, we compare STD-matching with three existing algorithms, the ST-matching algorithm, the stMM algorithm, and the HMM-RCM algorithm, using a real data set. The experiment results show that our STD-matching algorithm outperforms the three existing algorithms in terms of matching accuracy.**

***Index Terms*—Map Matching, GPS Data Analysis, Location-based Services**

## I. Introduction

With the popularity of mobile devices embedded with GPS (e.g., smartphones, personal navigation devices), a large number of GPS trajectory data are able to be collected by these devices nowadays. Because of GPS positioning errors, there may be inaccurate results if these trajectory data are used directly. Therefore, the map-matching technique, which is a process of associating a sequence of positions with the road network in a digital map, is considered as a basic preprocessing step for many applications.

There are several factors affecting the accuracy of map-matching algorithms. First, the sampling rate of GPS data has a great influence on matching accuracy. With high-sampling-rate GPS trajectories, more sampling points are given to form the trajectory. Therefore, highly accurate matching results can be produced. However, because of the strict limits of batteries, hand-held devices embedded with GPS are unable to collect high-sampling-rate data for a long period of time. Therefore, matching the low-sampling-rate GPS trajectories on maps has become an important issue. However, when dealing with low-sampling-rate GPS data which fetches a location every minute or even more, the location uncertainty increases as the sampling rate reduces. This uncertainty makes matching a low-sampling-rate trajectory a challenging issue. The second factor is the complexity of the road network, because ambiguous cases of matching are more likely to occur (e.g., ambiguity between a highway and general road when they run parallel to each other).

Although the issues of low-sampling-rate trajectories and complex road networks have been studied in [9] using spatial and temporal features, the proposed method, called ST-matching, still has some problems. First, the ST-matching algorithm cannot handle the matching problem at the beginning of the GPS trajectory. It only uses distance information to address this problem. Second, ST-matching cannot handle the GPS points around the junctions, sometimes matching to a wrong road segment. Third, when a trajectory consists of too many GPS points, the system may incur significant computation time to match the trajectory. In this paper, we propose a novel map-matching algorithm that considers the spatial, temporal, and directional features. The directional analysis function using real-time moving directional data is introduced. With these data, we can solve the matching problem, in particularly at junctions and at the beginning of the GPS trajectory. Then, we perform GPS clustering, GPS smoothing, and the $A^*$ shortest path algorithms to reduce the computational cost.

## II. Related Work

To solve the map-matching problem, a tremendous number of map-matching algorithms have been proposed lately. In the early stage, the most intuitive idea was to consider the closeness and the shape of the road segments. The basic approach, called point-to-point matching, was proposed in [1] which matches each GPS point to the nearest node in the road network. On the other hand, point-to-curve matching [16] finds the nearest point along the road segment to match with. Phuyal et al. proposed curve-to-curve matching [12]. Their algorithm identifies the candidate nodes with point-to-point matching. Next, it constructs the curve with the GPS trajectory and the curve with the candidate road segments corresponding to the road network, and compares the distance between two curves to find the most similar and the closest road segment as a best match. The approaches proposed in [5], [14] make use of the closeness and shape of the road segments as well as the connectivity of the road network. In [5], a weighted topological algorithm using the topology of the road network and the observation position of the vehicle was proposed,

978-1-5386-5862-8/17 $31.00 © 2017 IEEE
DOI 10.1109/SC2.2017.52

while in [14], Quddus et al. developed an enhanced topological map-matching algorithm based on the road network geometry to derive navigation data.

Newson et al. [11] proposed a map-matching algorithm based on HMM. In their paper, emission probabilities which model GPS noise as zero-mean Gaussian were utilized, while transition probabilities were modelled as this distance difference probability (i.e., the distance difference between GPS displacement and the match route distance). The authors found that the correct pairs of matched points typically result in a very short route. Finally, the Viterbi algorithm was used to compute the optimal path. To simplify the complexity of the HMM model, in [10], Mohamed et al. proposed an algorithm which uses three filters (i.e., a speed filter, a direction filter, and an $\alpha$-trimmed mean filter) to reduce the candidate sets for improving the efficiency of the map-matching process. In [6], Han et al. combined the road segments to reduce the capacity of the road network, and used a Hilbert R-tree to index the road network to improve the running time of searching the road network.

Wang et al. [15] proposed an algorithm based on the Ski-rental model, called Eddy, which includes solid error and delay-bound analysis. In different environments, this algorithm considers the tradeoff between accuracy and latency. Therefore, the algorithm dynamically changes the size of the sliding window to retrieve the best match from the candidate graph. In [7], He et al. introduced the confidence points (i.e., the GPS point which has only one candidate road segment) and the Maximum Delay Constraint Dynamic Time Window (MDCDTW) (i.e., a sliding window with a time constraint) to reduce the computation time. In [2], Goh et al. proposed an online map-matching algorithm based on HMM, which uses a variable sliding window (VSW) and a precomputed SVM-training probability function to improve the running time. AntMapper was proposed in [4], which considers both local topological information and global shape similarity measures. The main concept is to use the ant colony optimization algorithm to find the best path that matches the GPS trajectory on the map.

## III. The STD-Matching Algorithm

In this section, we outline an overview of our map-matching algorithm. In the following sections, we describe each component of our STD-Matching algorithm in detail.

### A. Candidate Set Extraction

Searching a matched road segment is time consuming when a road network is large. Hence, we retrieve a set of candidate points for each GPS point before finding the matched answers. Given a GPS trajectory $T : p_1 \to p_2 \to \cdots \to p_n$, we first search for a set of candidate road segments $e_i^j$ for GPS point $p_i$ in $T$ within radius $r$, where $1 \leq i \leq n$ and $j$ is an index of candidate road segments. Next, we compute the candidate points which have the minimum distance from the candidate road segments $e_i^j$ to $p_i$ with projection. The candidate points are chosen by the following principles: if the projection of $p_i$ onto the candidate road segment is located between its start point and end point, we choose the projection point as the candidate point; otherwise, we choose the endpoint that is nearest to $p_i$. We use $e_i^j$ and $c_i^j$ to respectively denote the $j$-th candidate road segment and the candidate point of $p_i$.

### B. STD Analysis

In order to choose the best matched answer from these candidate paths, we consider spatial, temporal, and directional information to model the score function for these candidate paths. With these three analyses, the output of this component is a candidate graph, and each node and edge of the graph is assigned a score.

*1) Spatial Analysis:* In spatial analysis, we consider both the geometrical and topological information of the road network to determine the matching probability of the candidate points found in the previous step. For the geometrical information, we use *observation probability*; for the topological information, we use *transition probability*.

If the candidate point is far from the measured GPS point, the likelihood of the measured GPS point matching to the candidate point is lower. Therefore, the observation probability can be reasonably formalized as a zero-mean normal distribution $N(\mu, \sigma^2)$ with a standard deviation $\sigma$. Then, we define observation probability $O(c_i^j)$ as the following equation:

$$O(c_i^j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d_i^j - \mu)^2}{2\sigma^2}} \tag{1}$$

, where $d_i^j = distance(p_i, c_i^j)$ is the distance between $p_i$ and $c_i^j$.

We also consider the topological information of the road network to increase the accuracy of the matching results. For example in Figure 1, the thin line is a local road segment and the thick line is a highway road segment. These two types of road segments are not connected to each other. $p_{i-1} \to p_i \to p_{i+1}$, which is a GPS trajectory of a user needs to be matched on the map. If we only consider the observation probability, $p_i$ would be matched to $c_i^1$. However, a vehicle is unlikely to take a long distance path from a highway to a local road and then go back to the highway later. Therefore, given $p_{i-1}$ and $p_{i+1}$ that are matched to the highway, $p_i$ should be matched to $c_i^2$ rather to $c_i^1$.

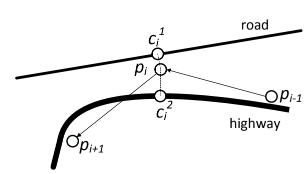


Fig. 1. An example of using transition probability.

To solve the problem in Figure 1, we calculate the shortest path between two neighboring candidate points $c_{i-1}^t$ and $c_i^s$. With the distance ratio of $distance(p_{i-1}, p_i)$ and the shortest path from $c_{i-1}^t$ to $c_i^s$, we formalize the transition probability as the following equation:

$$T(c_{i-1}^t, c_i^s) = \frac{distance(p_{i-1}, p_i)}{SP(c_{i-1}^t, c_i^s)} \quad (2)$$

, where $SP(c_{i-1}^t, c_i^s)$ is the length of the shortest path from $c_{i-1}^t$ to $c_i^s$. In this paper, we perform the $A^*$ search algorithm [3] to reduce the computational cost of searching for the shortest path between two neighboring candidate points.

By integrating Equations (1) and (2), we define the **spatial analysis function** $F_s(c_{i-1}^t \to c_i^s)$ as the following equation:

$$F_s(c_{i-1}^t \to c_i^s) = O(c_i^s) * T(c_{i-1}^t \to c_i^s), \quad 2 \le i \le n \quad (3)$$

, where $c_{i-1}^t$ and $c_i^s$ are two neighboring candidate points of GPS points $p_{i-1}$ and $p_i$. When dealing with the first point $p_1$ of GPS trajectory $T$, Equation (3) would be changed to $F_s(c_1^s) = O(c_i^s)$. Equation (3) computes the probability of a vehicle moving from $c_{i-1}^t$ to $c_i^s$ based on the geometrical and topological information. After spatial analysis, we can obtain a set of candidate paths $c_{i-1}^t \to c_i^s$ between any two adjacent GPS points $p_{i-1}$ and $p_i$, and finally, each candidate path is given a spatial analysis score by Equation (3).

*2) Temporal Analysis:* In spite of the spatial analysis considering the geometrical and topological information, there are some situations that spatial analysis could not handle. For example, when matching a GPS point $p_i$ which is close to a free way and local road segments. In this situation, the spatial analysis function would produce the same probability. However, the travelling speeds of these two types of road segment are different. If a vehicle is moving at a speed of 90 km/h, we would match it to the highway. In general, the average travel speed is similar to the road speed constraint. The observation of this behavior can be used to identify the candidate path with the most similar speed condition during the time interval. Therefore, we take the speed constraints of the road segments into consideration to solve this problem. Given two candidate points $c_{i-1}^t$ and $c_i^s$ of the GPS points $p_{i-1}$ and $p_i$, the shortest path from $c_{i-1}^t$ to $c_i^s$ can be represented as a list of road segments $[r_1', r_2', \cdots, r_k']$. First, we compute the average speed of the shortest path $\overline{v}_{c_{i-1}^t \to c_i^s}$ from $c_{i-1}^t$ to $c_i^s$. Additionally, each road segment $r_u'$ in the road network is assigned a speed constraint $r_u'.v$. Then, we can compute the similarity between the average speed and the speed constraint of the shortest path based on cosine similarity. Finally, the **temporal analysis function** can be formulated as the following equation:

$$F_t(c_{i-1}^t \to c_i^s) = \frac{\Sigma_{u=1}^k (r_u'.v \times \overline{v}_{c_{i-1}^t \to c_i^s})}{\sqrt{\Sigma_{u=1}^k (r_u'.v)^2} \times \sqrt{\Sigma_{u=1}^k \overline{v}^2_{c_{i-1}^t \to c_i^s}}} \quad (4)$$

, where $2 \le i \le n$. $c_{i-1}^t$ and $c_i^s$ are the two neighboring candidate points of GPS points $p_{i-1}$ and $p_i$.

*3) Directional Analysis:* The direction of two adjacent GPS points linking together does not represent the true moving direction of the GPS points. Therefore, we collect the real-time moving direction $p_i.b$, which is the acceleration direction of the GPS point $p_i$ at timestamp $p_i.t$, from the magnetic field sensor and accelerometer which are embedded in the smartphone. As a result, we are able to compute the similarity between the real-time moving direction and the road segment direction. We further compute the angular difference between the real-time moving direction of a GPS point and the angle of a candidate road segment. For example, in Figure 2, when considering just the spatial and temporal analysis function alone, it is very likely to match the trajectory to the path $r_1 \to r_2$. However, if we calculate the angular difference $\Delta\theta(c_i^j)$ between the real-time moving direction $p_i.b$ and the angle of road segment $\theta_{e_i^j}$, we would match the trajectory to the path $r_1 \to r_3 \to r_4$.

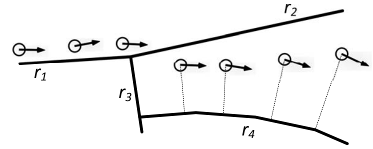


Fig. 2. An illustration of directional analysis.

In order to normalize the angular difference, we exploit normal distribution $N(\mu, \sigma^2)$ with a standard deviation $\sigma$. The normal distribution indicates how likely it is that a GPS point $p_i$ can be matched to a candidate point $c_i^j$ with directional information. Formally, we define the **directional analysis function** as the following equation:

$$F_b(c_i^j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\Delta\theta(c_i^j) - \mu)^2}{2\sigma^2}} \quad (5)$$

### C. Matched Answer Computation

In this section, we combine the spatial, temporal, and directional analysis that have been described in the previous section to retrieve the best matched answers for GPS trajectories. After the candidate set is extracted, we generate a candidate graph with the candidate points $c_i^s$ and candidate paths $c_{i-1}^t \to c_i^s$. After STD-analysis, each candidate point $c_i^s$ is assigned a score based on the observation probability $O(c_i^s)$ and the directional analysis function $F_b(c_i^s)$. Each candidate path $c_{i-1}^t \to c_i^s$ is assigned a score based on the transition probability $T(c_{i-1}^t \to c_i^s)$ and the temporal analysis function $F_t(c_{i-1}^t \to c_i^s)$. Finally, we combine the Equations (3), (4) and (5) to form the **STD-function** (defined in Equation 6) to find the matched answer for candidate path $c_{i-1}^t \to c_i^s$. The overall score of a matched path $F(P)$ can be presented as $\Sigma_{i=2}^n F(c_{i-1}^t \to c_i^s)$. The matched path with the highest overall score is regarded as the best matched answer.

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

## IV. Experiments

Our approach is compared with three existing algorithms: the ST-matching algorithm [9], the stMM algorithm [13], and the HMM-RCM algorithm [8]. In our experiment, we utilize the road network of Taipei city obtained from Open Street Map as shown. This road network contains 200,236 vertices and 90,709 road segments. We collected real-world trajectories using a data collector running on an Android system. The sampling interval of the GPS trajectory is in the range from one second to two minutes. In the pre-processing phase, we remove the GPS outliers based on the travelling speed constraint of $v_c = 50$ m/s (meter per second). We set $d_c = 4$ m (meter) as the cluster distance threshold, and set the slope difference constraint $s_d = 0.01$ to smooth the trajectory. In our approach, we search the candidate set for a GPS point within radius $r = 200$ m, but only choose $k$ candidate points for a GPS point. For observation probability, we use a normal distribution with $\mu = 0$ and $\sigma = 20$ m. For the directional analysis function, we use a normal distribution with $\mu = 0$ and $\sigma = 30°$.

Figure 3 shows the accuracy with respect to the different sampling intervals. We can see that our approach outperforms the three existing algorithms. When matching trajectory data with a 1-second sampling interval, the HMM-RCM algorithm performs better than the stMM and ST-matching algorithm. As the sampling interval increases, the accuracy of the stMM algorithm drops dramatically, while our approach and the ST-matching algorithm have more stable performance. We also notice that the average matching accuracy of our approach is still higher than that of the ST-matching algorithm when the sampling interval is over 80 seconds. This implies that the directional analysis improves the matching accuracy when dealing with low-sampling-rate trajectory data.

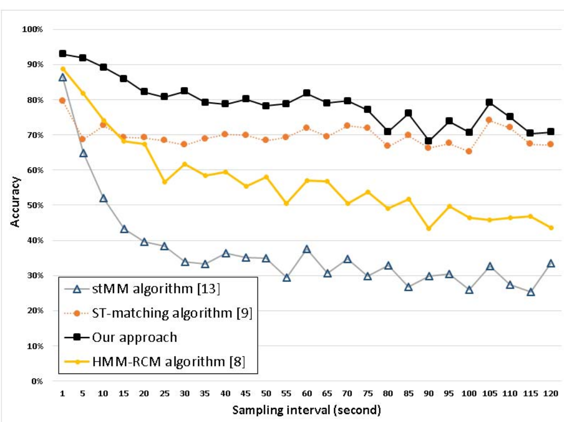


Fig. 3. Accuracy w.r.t sampling intervals.

## V. Conclusions

In this paper, we deal with the problem of map matching for low-sampling-rate GPS trajectories. A map-matching algorithm called the STD-matching algorithm is proposed and analyzed in this paper. The algorithm employs the real-time moving direction with the directional analysis function to reflect the influence of a user's true movement over the GPS trajectories. We construct the STD-matching algorithm which considers all of the spatial, temporal, and directional information. Furthermore, we reduce the computational cost by conducting GPS clustering, GPS smoothing, and the $A^*$ shortest path search algorithms. We finally evaluate our approach with three existing algorithms. The experiment results show that our STD-matching algorithm outperforms the ST-matching algorithm, the stMM algorithm, and the HMM-RCM algorithm in terms of matching accuracy for both high-sampling-rate and low-sampling-rate trajectories and concludes the utility of our novel approach.

## References


[1] David Bernstein and Alain Kornhauser. An introduction to map matching for personal navigation assistants. 1998.

[2] CY Goh, J Dauwels, N Mitrovic, MT Asif, A Oran, and P Jaillet. Online map-matching based on hidden markov model for real-time traffic sensing applications. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 776–781. IEEE, 2012.

[3] Andrew V Goldberg and Chris Harrelson. Computing the shortest path: A search meets graph theory. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 156–165. Society for Industrial and Applied Mathematics, 2005.

[4] Y. J. Gong, E. Chen, X. Zhang, L. M. Ni, and J. Zhang. Antmapper: An ant colony-based map matching approach for trajectory-based applications. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–12, 2017.

[5] Joshua S Greenfeld. Matching gps observations to locations on a digital map. In *Transportation Research Board 81st Annual Meeting*, 2002.

[6] Jingyu Han, Xiong Fu, Linfeng Liu, and Dawei Jiang. Online map matching by indexing approximate road segments. In *Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on*, pages 299–303. IEEE, 2011.

[7] Zhao-cheng He, She Xi-wei, Li-jian Zhuang, and Pei-lin Nie. On-line map-matching framework for floating car data with low sampling rate in urban road networks. *Intelligent Transport Systems, IET*, 7(4):404–414, 2013.

[8] G. R. Jagadeesh and T. Srikanthan. Online map-matching of noisy and sparse location data with hidden markov and route choice models. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–12, 2017.

[9] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 352–361. ACM, 2009.

[10] Reham Mohamed, Heba Aly, and Moustafa Youssef. Accurate and efficient map matching for challenging environments. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 401–404. ACM, 2014.

[11] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343. ACM, 2009.

[12] Bishnu P. Phuyal. Method and use of aggregated dead reckoning sensor and gps data for map matching. In *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*, pages 430–437, September 2002.

[13] Mohammed Quddus and Simon Washington. Shortest path and vehicle trajectory aided map-matching for low frequency gps data. *Transportation Research Part C: Emerging Technologies*, 55:328–339, 2015.

[14] Mohammed A Quddus, Washington Yotto Ochieng, Lin Zhao, and Robert B Noland. A general map matching algorithm for transport telematics applications. *GPS solutions*, 7(3):157–167, 2003.

[15] Guanfeng Wang and Roger Zimmermann. Eddy: an error-bounded delay-bounded real-time map matching algorithm using hmm and online viterbi decoder. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 33–42. ACM, 2014.

[16] Christopher E White, David Bernstein, and Alain L Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, 8(1):91–108, 2000.