International Journal of Digital Earth

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tjde20

A Novel Map Matching Method Based on Improved Hidden Markov and Conditional Random Fields Model

Wei Li, Youliang Chen, Shiteng Wang, Hongchong Li & Qin Fan

To cite this article: Wei Li, Youliang Chen, Shiteng Wang, Hongchong Li & Qin Fan (2024) A Novel Map Matching Method Based on Improved Hidden Markov and Conditional Random Fields Model, International Journal of Digital Earth, 17:1, 2328366, DOI: 10.1080/17538947.2024.2328366

To link to this article: https://doi.org/10.1080/17538947.2024.2328366

Published online: 12 Mar 2024.

INTERNATIONAL JOURNAL OF DIGITAL EARTH
2024, VOL. 17, NO. 1, 2328366
https://doi.org/10.1080/17538947.2024.2328366

OPEN ACCESS

A Novel Map Matching Method Based on Improved Hidden Markov and Conditional Random Fields Model

Wei Li[a], Youliang Chen[a], Shiteng Wang[b], Hongchong Li[a] and Qin Fan[a]

[a]School of Civil and Surveying & Mapping Engineering, Jiangxi University of Science and Technology, Ganzhou, People's Republic of China; [b]Ganzhou Ruida Survey Planning Co., Ltd, Ganzhou, People's Republic of China

ABSTRACT

Aiming at the inherent "labeling bias" problem of Hidden Markov Models (HMMs) in the process of low-frequency sampling rate GPS trajectory map matching, this paper proposes a map matching method that combines HMMs and Conditional Random Field Models (CRFs). Various features including vehicle driving direction, direction change between candidate roads, shortest path, and spatial localization accuracy of trajectory points are integrated to optimize the state transfer process in the HMM model. The Viterbi algorithm was then employed to calculate the joint probability values of all candidate paths and the path with maximum probability value was selected as the matching path. The proposed map matching method was validated using publicly available Global Positioning System (GPS) vehicle trajectories from Guangzhou, Dongguan, and field-collected vehicle trajectory data from Ganzhou, China. The results demonstrated that the proposed method achieves high matching accuracy while maintaining efficient matching efficiency. The percentage of correct matching of GPS sampling points remains above 95%, and the average calculation time for each candidate road segment is kept within 50 ms. These advancements can greatly benefit location-based services that rely on map matching technology.

ARTICLE HISTORY
Received 7 December 2023
Accepted 4 March 2024

KEYWORDS
Map matching; hidden markov model (HMM); conditional random fields (CRFs); floating vehicle; GPS trajectory

1. Introduction

In recent years, tracking devices equipped with global navigation satellite system (GNSS) receivers have been used in various types of vehicles. These sensor devices can obtain a vehicle's travel speed, direction, and location points. Further, this location information could be combined to form complete trajectory data, a series of positioning points. In intelligent transportation systems, these trajectory data are usually integrated with Geographic Information System (GIS) to provide various services for individual or corporate users, including navigation (Preiss et al. 2018), vehicle tracking (Jonietz et al. 2017), and route pushing (Harrison et al. 2014). Therefore, an in-vehicle positioning system plays an essential role in the construction of an intelligent transportation system. A series of Location-Based Service (LBS) applications such as vehicle navigation (White, Bernstein, and Kornhauser 2000), route recommendation (Sama et al. 2018), travel time estimation (Wang et al. 2019), and travel trajectory anomaly detection (Santhosh, Dogra, and Roy 2021) are inseparable from the support of the positioning system. However, these LBS applications are influenced remarkably by the accuracy of the coordinates derived from Global Position System (GPS) sensors and consequently

CONTACT Youliang Chen chenyouliang@jxust.edu.cn School of Civil and Surveying & Mapping Engineering, Jiangxi University of Science and Technology, No. 1958, Hakka Avenue, Ganzhou 341000, People's Republic of China

the matching of that coordinate to real electric road map. The process of matching these deviated trajectories with real road maps is called map matching or road network matching (Zhang et al. 2021).

Since the concept of map matching was introduced in the 1980s (Zheng et al. 2012), many studies on map matching have been done. These studies can be classified according to the map matching methods, the data sources used in the algorithms, and the GNSS trajectory and road network information considered in the matching process (Quddus, Ochieng, and Nol 2007; Wei et al. 2013). They are mainly classified into: geometry-based algorithms (White, Bernstein, and Kornhauser 2000) (such as point to point, curve to curve and point to curve), topological algorithms (Alejandra Blazquez et al. 2018; Blazquez and Vonderohe 2005; Zhao et al. 2018), probabilistic algorithms (Bierlaire, Chen, and Newman 2013; Xu, Zhang, and Chen 2020), and advanced map-matching algorithms (Karamete, Adhami, and Glaser 2021; Sarvrood, Liu, and Gao 2018; Wang, Metzner, and Schwieger 2020; Zhao et al. 2018). Geometry-based algorithms focus on the distance between the GPS point and the candidate roads. They also consider the similarity between the trajectory and candidate roads.

White, Bernstein, and Kornhauser (2000) proposed several geometry-based algorithms. The first algorithm is very similar to the point-to-point matching method, in which the algorithm searches for road arcs close to the GPS location points, and then projecting the GPS points into the nearest arc. The second algorithm additionally considers course information based on the previous algorithm. The third algorithm is a variant of the second method, which uses the topological information of the road network, i.e. the reachability between road segments. Given a set of arcs, the unreachable arcs are excluded by determining the reachability between GPS points and each segment of arcs, which increases the matching performance of the algorithm to a certain extent. All three of the aforementioned matching algorithms have their own limitations. The first algorithm does not take into account the historical information of the trajectory points. The second algorithm increases the matching time when the heading of the GPS trajectory deviates significantly from the heading of the arc. As for the third algorithm, it suffers from an unstable matching process in which the mismatch of the previous point may also lead to the mismatch of the subsequent series of points as well. Nassreddine, Abdallah, and Denoeux (2008) proposed an advanced map-matching algorithm based on belief function theory. The algorithm manages the bounded vehicle location output by combining interval data and rectangular road maps in the framework of confidence functions. This approach has a stable output when applied to roadway intersections and can handle multi-hypothesis scenarios for intersections. HMM is an advanced map matching algorithm released by Microsoft in 2009 (Newson and Krumm 2009). The basic idea is to decompose the probability of a GPS point of a candidate road into an observation probability and a transfer probability. HMM has achieved good results in map matching in many scenarios and is becoming increasingly popular. In recent years, map matching methods based on deep learning and cloud computing (Huang et al. 2015) have been gradually proposed. For example: RNN-based map matching schemes (Shen, Du, and Zhao 2020), Deep MM (Zhao et al. 2019), and machine learning-based map matching (Hashemi 2017). However, the application of deep learning methods in the field of map matching is just in the initial stage. In addition, to train a model, a large amount of labeled data is required, and labeled data is difficult to obtain, and it is equally challenging to take a manual approach to label the data.

Although as an essential part of establishing current and future intelligent transportation systems, scholars proposed different models, from the initial and simplest geometric map matching methods to the most common advanced map matching methods, most of the proposed map matching methods have difficulty in balancing the relation between matching accuracy and matching time efficiency. Zhao et al. (2018) proposed an advanced topological map matching algorithm based on D-S theory. The algorithm changes the distance geometric features into two topological relations, road speed limit and spatial correlation, and calculates the weights of each factor based on BPA, which in turn calculates the fusion result. However, the model does not take into account the fact that the GPS is traveling at low speeds when the speed factor is introduced, which leads to inaccuracies in the acquired positioning information. Singh, Singh, and Sehra (2020) proposed a post-processing Genetically Inspired Map Matching (GIMM) algorithm. The fitness function used utilizes Bucket Dijkstra and Fast Dynamic Time Wrapping (FDTW) algorithms to present GPS information on digital maps with low memory requirements and relatively easy computation. Singh et al. (2019)compared geometric, topological and Kalman filter based map matching algorithms on the same dataset. The results show that the performance of the Kalman filtering algorithm provides better results compared to the geometric and topological algorithms. Luo, Chen, and Xv (2017) constructed an enhanced Hidden Markov Model Road network matching algorithm based on GPS and cell phone positioning data by constructing the topological matrix of road network segments and refining the quadtree data structure. They claimed that they solved the problem of poor matching accuracy of the algorithm when dealing with low sampling rate data. Che et al. (2018) proposed an enhanced Hidden Markov Model based on the existing Floating Car Data (FCD), which improves the matching accuracy of the algorithm at complex road segments by incorporating historical FCD information Based on the traditional Markov Model.

HMM demonstrates good processing and improvement capabilities in the map matching problem, and shows a large advantage among the existing map matching algorithms. However, the 'labeling bias' problem arising from the normalization of local transfer probabilities in HMM models cannot be ignored. To address this issue, we introduce a novel hybrid trajectory algorithm (HMM-CRFs), incorporating a conditional random field to estimate weight coefficients based on eigenfunctions, thereby leveraging more global information. Subsequently, this coefficient is added to the state transfer probability model to improve the matching accuracy and overall matching efficiency of complex road sections. Our algorithm enhances map matching accuracy and efficiency by considering the spatiotemporal context information of GPS trajectory points and the topological relationship between road networks. The main research contributions of this paper are as follows:

- A map matching method based on HMM-CRFs that integrates both spatial and temporal information is proposed. The method considers distance, direction, and spatial reachability in the process of calculating observation probability and transition probability based on HMM to improve the correct matching rate of the algorithm at complex intersections.
- Using the Geohash algorithm to construct a road network grid index, setting the radius of the error circle, reducing the computational complexity of the map matching algorithm on candidate road segments during operation, and ensure the rate of the map matching algorithm.
- Based on the traditional HMM model, the weight adjustment coefficient is introduced to solve the problem of low matching accuracy caused by the 'annotation bias' problem in the low sampling rate GPS trajectory of the traditional HMM model.

2. Hybrid road network matching model based on HMM-CRFs

This section briefly describes the methodology for obtaining road sets and constructing hybrid models of HMM-CRFs, explains how to use the Viterbi algorithm to implement trajectory transformations as well as selecting appropriate evaluation metrics to validate the accuracy of the hybrid models.

2.1. Related definition and theoretical foundations

The relevant concepts and definitions involved in the map matching process are described as follows:

Road Segment: Two adjacent nodes are connected to form a road segment E, which contains information about the attributes of the road segment, such as road segment number E.ID, road segment name E. Name, road segment length E. Length and road segment speed limit E. Speed, etc.

Road network: The connection structure in the urban road network can be represented by a directed graph $G(V, E)$. The edges E in the directed graph represent each road segment in the road network, the nodes V represent the connection nodes between the road segments, and a complete road network structure is composed of many sets of road segment nodes and sets of road segments.

GPS trajectory: Any GPS trajectory T $= \{P^{i}_{GPS} | i = 1, 2, 3, \cdots, n\}$ is connected by a series of GPS positioning points (sampling points) with time-stamped markers, each containing information such as GPS data reception time, latitude and longitude, speed, and vehicle travel direction.

GPS trajectory dataset: A vehicle equipped with a GPS receiver driving on the city road will generate a series of GPS points, and the GPS points in a period can be connected to a GPS track, and all the GPS tracks constitute a set $L_T = \{T_i | i = 1, 2, 3, \cdots, m\}$ which is called a GPS track log record.

GPS candidate points: A vertical projection is used to project the GPS track points onto the road segments within the error circle. All the projected points obtained constitute the set of candidate points C $= \{C_i | i = 1, 2, 3, \cdots, N\}$.

2.2. Candidate road set acquisition

In the HMM-based map matching method, the hidden state sequence is a set $\Phi = \{e_i^1, e_i^2, e_i^3, \cdots, e_i^N\}$ of all candidate road segments centered on the trajectory points. In general, all roads in the study area should be selected as the candidate set, but this will greatly prolong the computation time of the algorithm and consume a large amount of computational cost. Therefore, the candidate road set for trajectory data must be reduced (Jagadeesh and Srikanthan 2017), in which the topological connectivity between the road segments is ensured, and the real paths traveled by vehicles are included as completely as possible.

In order to reduce the computational burden of map matching algorithms on candidate road segments during operation and improve the efficiency of the algorithm, a longitude and latitude encoding method, Geohash algorithm, is used to grid the road network and construct a road network grid index. Establishing a grid index can enable the calculation of candidate road segments to obtain the best matching road segment with relatively few comparisons. The basic idea of Geohash algorithm is that the earth can be divided into four parts, namely $[-90°, 0°)$,$(0°, 90°]$,$[-180°, 0°)$ and $(0°, 180°]$, by taking the Prime meridian and equator as the boundary. In the process of searching for candidate road segments, the length of the Geohash code is set to 8 bits, and the accuracy of grid partitioning is determined to be within 100 meters. First, calculate the projection coordinate points of each GPS track point on each candidate road segment. By searching the distance from the current GPS track point to each candidate point (projection point), the distance is determined by comparing the similarity of Geohash codes. The more matching digits between the current GPS track point and the candidate point code, the closer the GPS track point is to the candidate point. The Geohash algorithm can be divided into the following three steps:

Step 1: Converts latitude and longitude to binary code. According to the encoding idea of Geohash, the latitude and longitude will be divided step by step until the number of divisions in the middle of the division is very close to the latitude and longitude data, then stop the division, according to the division of the interval using binary 0 and 1 encoding.

Step 2: Combine the longitude and latitude codes into one whole binary code. The longitude code occupies an even number of digits and the latitude code occupies an odd number of digits, combining latitude and longitude.

Step 3: Encoding of binary numbers based on the idea of Base32 encoding. Starting from bit 0, every 5-bit binary code corresponds to a Base32 code, and so on in sequence, and finally the encoded values are combined into a string.

The error circle is then utilized to select the candidate road segments within the study area. Figure 1 represents the method of selecting the set of candidate road sections using error circles. In Figure 1, the center point P_i is a GPS positioning point obtained from the vehicle's driving track. The set Geohash encoding length is 8 bits, i.e. the precision of the grid is controlled within 100 m, since during the grid search, 8 surrounding grids are searched. The precision of the 9 grids is about 300 × 300 m, and the grids are extended by 50 m for each direction separately as the radius of the error circle. So, take the trajectory point P_i as the center and construct the error circle with $R = 200$ m as the radius. The road segment that falls within the error circle range is the trajectory point's candidate road segment set. The yellow line in the figure includes the road segment r_i^1, r_i^2, r_i^3, r_i^4, r_i^5 and r_i^6, the blue dot C_i^2, C_i^3, C_i^4, C_i^5 represent the projection point of the trajectory point P_i on the candidate road segment, also known as the candidate point. Although the road segments r_i^1 and r_i^6 are also candidate road segments, the projection point of the trajectory point P_i is on its extension line. At this time, the road segment nodes C_i^1 and C_i^6 closest to P_i are selected as candidate points. This calculation method is based on the matching idea of a geometric map matching the midpoint to a straight line and taking the closest point from the trajectory point to the road segment as the matching point.

2.3. Dilution of GPS track data

Due to factors such as sampling frequency and sampling length, the interval between the obtained vehicle trajectory data points is minimal, the data points are relatively dense, and the data points have the same information. To improve the efficiency of the algorithm, the original trajectory data is diluted using the Douglas-Peucker algorithm while ensuring that the trajectory does not change. The Douglas Pucker algorithm approximates polygons or curves as a series of points, reducing intermediate points through some processing steps. The algorithm was proposed by David Douglas and Thomas Peucker in 1973 (Jung et al. 2019) and was further refined and developed in the following decades. The main steps for running the Douglas-Pucker algorithm are as follows:

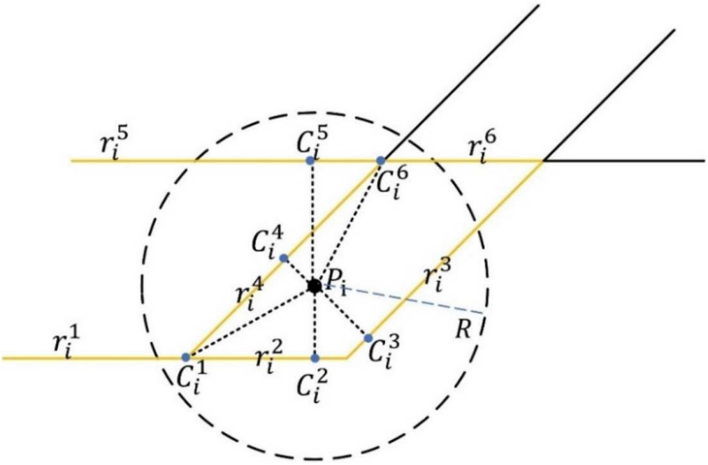

Figure 1. Error circle and candidate section set.

Step 1: Connect the starting and ending trajectory points in the vehicle trajectory sequence to form a straight line, called the chord of the curve, and set the threshold Dthreshold simultaneously.

Step 2: Search along the starting point of the trajectory until the ending point, obtain the trajectory point farthest from the line, and calculate the distance from the point to the line, recording the distance as Dmax.

Step 3: Compare the size of Dmax with the threshold Dthreshold. When Dmax < Dthreshold, remove all trajectory points in the middle of the line and use the line as the approximate trajectory curve. The processing of this trajectory segment ends.

Step 4: When Dmax > = Dthreshold, retain the trajectory point, use this point as the endpoint, divide the trajectory into left and right segments, repeat steps 1–3 until all points are traversed, and connect all trajectory points processed by the algorithm as approximate curves of the original trajectory.

2.4. HMM-CRFs hybrid model

2.4.1. Overview of the model

The construction process of the map matching method based on the hybrid model of HMM-CRFs is shown in Figure 2. The steps for constructing a hybrid model of HMM-CRFs are as follows:

(a) Data preprocessing for FCD and urban road network data.

In order to reduce the time complexity of the algorithm and improve the efficiency of the computation, before using the CRFs algorithm to compute the error cost adjustment weight factors, first, the GPS trajectory points were diluted using the Douglas-Peucker algorithm. Then, the spatiotemporal feature information in the GPS track points and urban road network data is extracted, and the set of candidate road sections is obtained based on the feature information of the GPS track points to avoid searching the whole urban road network in the process of map matching. This process uses GPS trajectory data and urban road network data. GPS trajectory data consists of structures such as Time, Longitude, Latitude, Speed and Num. Urban road network road network data consists of structures such as Road segment number, Longitude of the start of the road segment,

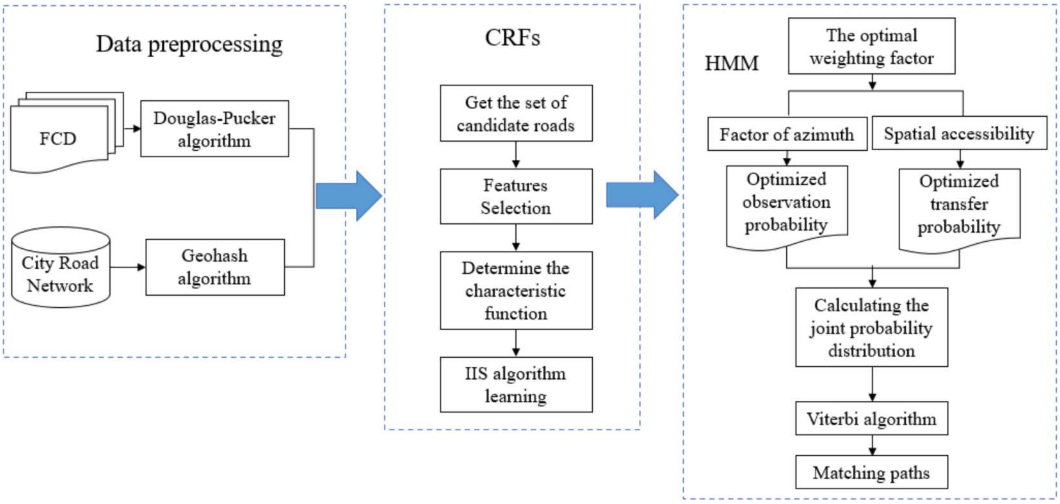
Data preprocessing
FCD
Douglas-Pucker algorithm
City Road Network
Geohash algorithm
CRFs
Get the set of candidate roads
Features Selection
Determine the characteristic function
IIS algorithm learning
HMM
The optimal weighting factor
Factor of azimuth
Spatial accessibility
Optimized observation probability
Optimized transfer probability
Calculating the joint probability distribution
Viterbi algorithm
Matching paths

Figure 2. Hybrid model framework of HMM-CRFs.

Latitude of the start of the road segment, Longitude of the endpoint of the road segment, Latitude of the endpoint of the road segment and Length of a road segment. The data are used for the following purposes.

(b) Computation of error cost adjustment weight factors using CRFs algorithm.

Based on the diluted GPS track point data and the corresponding candidate road sets, the spatio-temporal features of GPS track points are extracted. Construct the state feature function and the transition feature function to establish the posterior probability function of CRFs matching a specific path under the given GPS track. According to the posterior probability function, an improved iterative scaling (IIS) algorithm, a machine learning method, is used to estimate the optimal weight coefficient in the CRFs model. The error cost adjustment weight factors are then calculated. The task of the IIS algorithm is to calculate the gradient of the objective function in the process of parameter learning and then use the gradient to search for the optimal solution. The data structures used in this process mainly include Feature Functions, Feature Weights, Observation Sequence, Label Sequence, Transition Matrix, State Feature Vector and Global Features.

(c) Constructing the observation probability model and transfer probability model in the optimized HMM.

The cost-adjustment weight factors computed from the CRFs model are added to the observation probability model and the transfer probability model in the HMM to compute the optimized observation probability and transfer probability of the current hidden state. The main data structures used in this process include Set of States, Set of Observations, State Transition Probability Matrix, Observation Probability Matrix and Initial State Probability Distribution.

(d) Finding the optimal matching path using the Viterbi algorithm.

Based on the derived observation probabilities and transfer probabilities, the joint probability of the current hidden state is solved using the Viterbi algorithm. Record the previous state of the current hidden state, solve for the traveling path with the highest probability, and decode the optimal path for the GPS trajectory. The data structure of the Viterbi algorithm mainly consists of State Sets, Observation Sequence, State Transition Probability Matrix, Observation Probability Matrix, Initial State Probability Distribution, Viterbi Path Probability Table and Backtracking Table.

2.4.2. Optimized observation probability model

Observation probability is the probability value of producing a visible observation state under a hidden state sequence. In map matching, observation state probability is the probability of producing one of the trajectory points when a vehicle is driving on a certain road. The GPS receiver may return multiple trajectory location points. According to empirical knowledge, the closer the trajectory point is to the road, the higher the probability of falling on that road. If the spatial positioning accuracy of the vehicle trajectory point obeys Gaussian distribution, i.e. the distance between the trajectory point and the candidate road can be modeled by Gaussian distribution, then the conditional probability of the distance between the GPS observation points $o_i^{(t)}$ to the candidate's road $c_i^{(t)}$ can be expressed by the following equation.

$$P(c_i^{(t)}|o_i^{(t)}) = \frac{1}{\sigma_d\sqrt{2\pi}}\exp\left(-\frac{d(o_i^{(t)} \to c_i^{(t)})}{2\sigma_d^2}\right) \tag{1}$$

where σ_d^2 represents the variance of the GPS positioning error. Since the positioning accuracy of GPS is affected by the number and angle of satellites, multipath effects in the ground environment, and the Selective Availability (SA) measures of the U.S. government (Li, Jia, and Zhao 2020), the confidence level of the general vehicle positioning point at 100 m from the road is 95%. However, since May 2000, with the gradual elimination of SA measures, the horizontal positioning error of civilian GPS receivers has shrunk to about 10 m (Shen and Stopher 2014). Combined with the width of urban roads, which is usually around 20 m, half of the road width is taken, so the value range of σ_d is set between 10 and 20 $d(o_i^{(t)} \rightarrow c_i^{(t)})$ represents the vertical projection distance from the GPS observation point to the candidate road segment.

Only considering the distance between the GPS observation point and the candidate road segment obviously cannot meet the accuracy requirements of offline map matching, and additional restriction factors need to be added to improve the matching accuracy. In this paper, the factor of azimuth is added, i.e. the similarity between the vehicle's driving direction and the direction of the candidate road segment is considered. The direction is generally the angle between the driving direction of the vehicle and the due north direction {θ|θ ∈ [0, 360]}, and the conditional probability of the direction is also defined by the following normal distribution:

$$P(c_{angle}^{(t)}|o_{angle}^{(t)}) = \frac{1}{\sigma_{angle}\sqrt{2\pi}} exp\left(\frac{\theta(o_{angle}^{(t)}, c_{angle}^{(t)})}{2\sigma_{angle}^2}\right) \tag{2}$$

where σ_{angle}^2 represents the variance of the direct observation error, which is set in the range of [20,45] in this paper, $o_{angle}^{(t)}$ and $c_{angle}^{(t)}$ represent the GPS observation point and the positive The angle between the north direction and the road direction and the true north direction, $\theta(o_{angle}^{(t)}, c_{angle}^{(t)}) = |o_{angle}^{(t)} - c_{angle}^{(t)}|$, the geometric relationship between the angles is shown in Figure 3.

Generally, the direction information is used as a separate dimension, independent of the position information. Combining Equaitons (1) and (2), the observation probability of GPS observation points is expressed as:

$$P_{obs} = \mu \cdot P(c_i^{(t)}|o_i^{(t)})P(c_{angle}^{(t)}|o_{angle}^{(t)}) \tag{3}$$

where μ denotes the cost factor, which is used to adjust the effect of anomalies in the GPS trajectory on the performance and matching accuracy of the map-matching algorithm. The magnitude of the adjustment parameter will be found in the parameter estimation subsection.

2.4.3. Optimized state transition probability model

The state transition probability represents the probability of a vehicle moving from the candidate road segments corresponding to the current GPS track point to the candidate road segments corresponding to the next GPS track point. In calculating the transfer probability, two assumptions are considered in this paper: (1) GPS track is the positioning point generated by the vehicle driving on the real road, and the linkage of the GPS track should roughly conform to the shape of the road, that

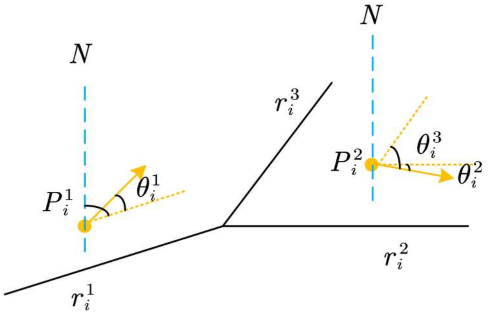

Figure 3. GPS observation points and the road segment direction of the angle representation.

is, the direction of the linkage of two adjacent GPS track points has greater similarity with the direction of the linkage of the corresponding candidate points; (2) Generally speaking, for the route selection preference of drivers, the possibility of choosing a road driving farther away from the path between two points is low, and they often tend to choose a closer route to drive.

For the hypothesis 1, the similarity in the direction of travel can be defined by the following equation:

$$P_\alpha = \left| \frac{\overrightarrow{o_{i-1}o_i} \cdot \overrightarrow{c_{i-1}c_i}}{|o_{i-1}o_i| \cdot |c_{i-1}c_i|} \right| \tag{4}$$

where $\overrightarrow{o_{i-1}o_i}$ denotes the direction vector between two adjacent GPS track points, $\overrightarrow{c_{i-1}c_i}$ denotes the direction vector between the GPS track points corresponding to the candidate points, and P_α denotes the conditional probability of direction shift between the candidate road segments, as represented in Figure 4.

In addition to considering the above assumption1, a spatial accessibility assumption needs to be added to prevent false matches; otherwise, as shown in Figure 3, when α_1 and α_2 are equal, then the directional transfer probabilities from the candidate point c_{i-1} to candidate points c_i^1 and c_i^2 are the same, and false matches may arise at this point. For assumption 2, Spatial accessibility, i.e. the conditional probability of the shortest path between neighboring candidate points can be defined by the following equation:

$$P(r_m, r_n) = \frac{d(o_{i-1} \to o_i)}{d_r(c_{i-1} \to c_i)} \tag{5}$$

where $d(o_{i-1} \to o_i)$ denotes the Euclidean distance between adjacent GPS observation points,-$d_r(c_{i-1} \to c_i)$ denotes the shortest path distance between GPS observation points corresponding to candidate points. In this study, the distance of the shortest path is solved by Dijkstra's algorithm, and Figure 5 represents the candidate section transfer under this assumption.

Combining Equations (4) and (5) one can calculate the joint conditional probability of transfer between candidate sections as:

$$P_{trans} = \omega \cdot P_\alpha \cdot P(r_m, r_n)\,. \tag{6}$$

where ω denotes the magnitude of the effect of directional transfer and spatial accessibility conditions on the conditional probability distribution of transfer between candidate road segments.

2.4.4. Model parameter estimation

Two error cost adjustment weight factors are added to the observation Statistical model and the state trsition Statistical model respectively. The calculation of the weight coefficients in the

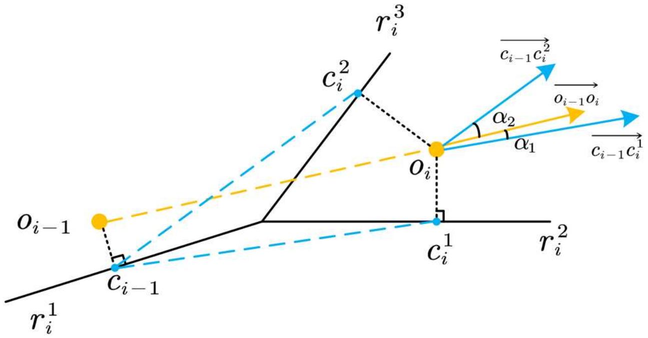
r_i^3
c_i^2
$\overrightarrow{c_{i-1}c_i^2}$
$\overrightarrow{o_{i-1}o_i}$
α_2
α_1
$\overrightarrow{c_{i-1}c_i^1}$
o_i
o_{i-1}
c_i^1
r_i^2
c_{i-1}
r_i^1

Figure 4. Directional shift between candidate sections.

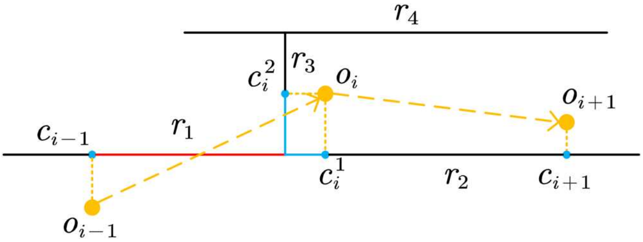

Figure 5. Spatial accessibility – shortest path transfer.

model can be determined by defining the posterior conditional probability functions of the CRFs and based on the extracted spatial–temporal features, the expressions of the CRFs can be established as follows:

$$\mathrm{P}(c_{i:N}|o_{i:N}) = exp\left(\sum_{t\le T}\mu\cdot\xi(c_i^{(t)}|o_i^{(t)})\cdot\varphi(c_{angle}^{(t)}|o_{angle}^{(t)}) + \sum_{t+1\le T}\lambda\cdot\psi_1(\alpha)\cdot\psi_2(r_m, r_n)\right) \quad (7)$$

where $\xi(c_i^{(t)}|o_i^{(t)})$ represents the projected distance feature function from the GPS observation point to the candidate road segment, $\varphi(c_{angle}^{(t)}|o_{angle}^{(t)})$ represents the feature function of the angle between the direction of the GPS track point and the direction of the candidate road segment, both of which belong to the state feature, and the corresponding weight coefficient is μ; $\psi_1(\alpha)$ represents the vehicle driving direction consistency feature function, $\psi_2(r_m, r_n)$ represents the shortest path distance feature function of spatial reachability. These two functions belong to the transfer feature function, and the corresponding weight coefficient is λ.

Equation (7) is simplified to facilitate subsequent calculations, and the feature functions and weight coefficients are first expressed in a unified form. Assuming that there are K_1 state features, K_2 transition features and the total number of features is K, where the number of K is 2, then:

$$\Psi_k(F_i) = \left\{\begin{array}{c} \tau_k(c^t|o^t),\ k = 1, 2, \cdots, K_1 \\ s_l(c^i \to c^{i+1}),\ k = K_1 + l, l = 1, 2, \cdots, K_2 \end{array}\right\} \quad (8)$$

$$\tau_k(c^t|o^t) = \xi(c_i^{(t)}|o_i^{(t)})\cdot\varphi(c_{angle}^{(t)}|o_{angle}^{(t)})$$

$$s_l(c^i \to c^{i+1}) = \psi_1(\alpha)\cdot\psi_2(r_m, r_n)$$

Denote byω_k the weight coefficient of the characteristic function $\Psi_k(F(i))$:

$$\omega_k = \left\{\begin{array}{c} \mu_k,\ k = 1, 2, \cdots, K_1 \\ \lambda_l,\ k = K_1 + l, l = 1, 2, \cdots, K_2 \end{array}\right. \quad (9)$$

If the vector of weight coefficients is denoted by ω, then the weight values can be expressed as $\omega = [\omega_1, \omega_2, \omega_3, \cdots, \omega_K]^T$. Considering F as a global feature vector, it can be expressed as:

$$F = [\tau_k(c^t|o^t), s_l(c^i \to c^{i+1})]^T \quad (10)$$

To infer the optimal value of the weight coefficients in the model, the method of finding the model parameters by maximizing the log-likelihood function of the training data. Given a joint posterior probability distribution $\tilde{P}(o^{1:T}, c^{1:T})$ from the training data set, combined with the normalized CRFs potential function, i.e. $\mathrm{P}(c_{i:N}|o_{i:N})$ the log-likelihood function of $\tilde{P}(o^{1:T}, c^{1:T})$ can be defined as:

$$\begin{aligned} L_{\tilde{P}}(\omega) &= L_{\tilde{P}}(\tilde{P}(o^{1:T}, c^{1:T})) \\ &= \sum_{i=1}^{T-1}\sum_{k=1}^{K} \omega_K \Psi_k(F_i) - \sum_{i=1}^{T-1} logZ_\omega(F_i) \end{aligned} \tag{11}$$

Given the feature function $\Psi(F)$, $\Psi(F)$ contains all the spatial–temporal features of the GPS trajectory points and the road network. In this study, a total of four spatial–temporal features, two-state features, and two transfer features are defined. In addition, the empirical distribution $\tilde{P}(o^{1:T}, c^{1:T})$ is given, and the optimal solution (the maximum value) $\hat{\omega}$ of the weight coefficient vector needs to be found such that $\hat{\omega} = argmaxL_{\tilde{P}}(\omega)$. The idea of the IIS algorithm is (Berger 1998), assuming that the weight vector in the model is $\omega = [\omega_{,11}, \omega_2, \omega_3, \cdots, \omega_K]^T$, when an increment δ is added to the weight vector, then the weight vector becomes $\omega + \delta = [\omega_1 + \delta_1, \omega_2 + \delta_2, \omega_3 + \delta_3, \cdots, \omega_K + \delta_K]^T$. In the process of algorithm iteration, by solving the state feature update equation and transfer feature update equation, thus obtaining $\delta = (\delta_1, \delta_2, \delta_3, \cdots, \delta_K)^T$.

The update equation for the state characteristics is:

$$E_{obs_{\tilde{P}}} = \sum_{(o^{1:T},c^{1:T})} (\tilde{P}(o^{1:T}, c^{1:T})) \times \sum_{k=1}^{K_1} \Psi_k(F_i) exp\left(\delta_k \sum_{k=1}^{K} \Psi_k(F_i)\right) \tag{12}$$

The update equation for the transfer characteristic is:

$$\begin{aligned} E_{trans_{\tilde{P}}} &= \sum_{(o^{1:T},c^{1:T})} (\tilde{P}(o^{1:T},\ c^{1:T})) \\ &\times \sum_{k=K_1+1}^{K} \Psi_k(F_i) exp\left(\delta_k \sum_{k=1}^{K} \Psi_k(F_i)\right) \end{aligned} \tag{13}$$

In Equations (12) and (13), K_1 denotes the number of observed state features. In this paper, spatial positioning accuracy and direction consistency are selected as observed state features and combined into one observed feature function $\tau_k(c^t|o^t)$ containing weights, so the value of K_1 s equal to 1. The transfer feature update equation contains two transfer features of vehicle travel direction consistency and shortest path distance, which are also combined into one transfer feature function $s_l(c^i \rightarrow c^{i+1})$. Containing weights, so the value of K is 2.

2.4.5. Viterbi algorithm for trajectory transformation

The Viterbi algorithm uses the idea of Dynamic programming to solve the prediction problem in the hidden Markov model, and to solve the driving path with the highest probability (Bloit and Rodet 2008), that is, to decode the optimal path of the GPS track. Introducing two variables o and m, o used to store the maximum probability corresponding to the path, m is used to store the node corresponding to the maximum probability of the path. The Viterbi algorithm calculation process is mainly divided into the following three steps:

Step 1: Calculate the observed probability of the first trajectory point corresponding to the GPS observation state at a certain time t:

$$O(i) = P_{obs}(c_i|o_{gps}) \tag{14}$$

Step 2: For the GPS observation point at time $t_k = t + t_0,\ \ t_0 > 0$, the maximum probability value of the corresponding path is obtained by the following equation:

$$o_{t_k}(i) = \max_{c_1, c_2, \cdots, c_i} P_{obs}(c_{t_k}|o_{gps}^{t_k})P_{trans}(c_{t_k}|c_{t_k-1}) \tag{15}$$

Step 3: At time t_k, the GPS trajectory 'true point' corresponding to the path with the highest probability among all candidate paths $\{c_1, c_2, \cdots, c_i\}$ is defined as follows:

$$m_{t_k}(i) = arg \max_{c_1, c_2, \cdots, c_i} P_{obs}(c_{t_k}|o_{gps}^{t_k})P_{trans}(c_{t_k}|c_{t_k-1}) \tag{16}$$

Finally, the candidate nodes corresponding to the maximum path probability are connected into a continuous sequence of time points, which are transformed into real vehicle trajectories through the Viterbi algorithm.

The Pseudocode of Viterbi algorithm to convert GPS track sequence data into real vehicle driving path is shown in Algorithm 1.

Algorithm 1 Viterbi algorithm for transforming GPS trajectory data

```
Input: $O = \{o_1, o_2, \ldots, o_N\}$ (Observation Space), $S = \{s_1, s_2, \ldots, s_K\}$ (State Space),
    $\prod = (\pi_1, \pi_2, \ldots, \pi_K)$ (Initial Probabilities), Tm (Transition Matrix), Em (Emission Matrix)
Output: best_path
Function Viterbi{O, S, $\prod$, Tm, Em): best_path
    trellis $\leftarrow$ matrix(length(S), length(O))
    pointers $\leftarrow$ matrix(length(S), length(O))
    for s in range(length(S)):
        trellis[s, 0] $\leftarrow$ $\prod$[s] $\cdot$ Em[s, O[0]]
    for o in range(length(O)):
        for s in range(length(S)):
            k $\leftarrow$ arg max(trellis[k, o – 1] $\cdot$ Tm[k, s] $\cdot$ Em[s, o] for k in range(lengeth(S)))
            trellis[s, o] $\leftarrow$ trellis[k, o – 1] $\cdot$ Tm[k, s] $\cdot$ Em[s, o]
            pointers[s, o] $\leftarrow$ k
  best_path $\leftarrow$ list()
  k $\leftarrow$ arg max(trellis[k, length(O) – 1] for k in range(length(S)))
  for o in range(length(O) – 1, -1, –1):
    best_path.insert(0, S[k])
    k $\leftarrow$ pointers[k, o]
    return best_path
END Function
```

The overall spatiotemporal complexity of the hybrid HMM-CRFs model mainly depends on the decoding step of using the Viterbi algorithm to realize the conversion of GPS track sequence data to real vehicle travel paths. The space and time complexity of the Viterbi algorithm is linear and proportional to the length of the observation sequence and the number of states. Therefore, the time complexity of the hybrid model is roughly $O(T \times N^2)$ and the space complexity is roughly $O(T \times N)$, where T is the length of the observation sequence and N is the number of hidden states.

2.5. Selection of evaluation indicators

Various evaluation indexes were calculated to evaluate our model performance and its matching feasibility. To evaluate the time efficiency of our proposed algorithm, the model running time under different conditions was investigated, while for the matching quality, i.e. the matching accuracy measurement, the proposed paper is an offline map matching method, which needs to consider the global matching quality (Pereira, Costa, and Pereira 2009). The following equation calculates the correct matching percentage (CMP) of the sampled points.

$$CMP = \frac{\#Number\ of\ \ correctly\ matched\ sample\ points}{\#\ Total\ number\ of\ \ sample\ points\ to\ be\ matched} \times 100\% \tag{17}$$

In the Equation (17), the total number of sample points to be matched refers to the GPS sampling points that need to be input into the model after preprocessing, not all the GPS sampling points in the original data.

3. Case study

This section briefly describes the sources and preprocessing results of the GPS trajectory data of the floating vehicle and explains the setup of the experimental environment.

3.1. Data source description

The GPS track data comes from the open datasets provided by Open ITS Open Alliance, and two open datasets provided by this organization are used in the experiment:

(1) The GPS track dataset was collected from the experimental road near Zhongshan University in Guangzhou, which is shown in Figure 6, with a total route length of about 18.6 km, including 12 different roads and ordinary roads.

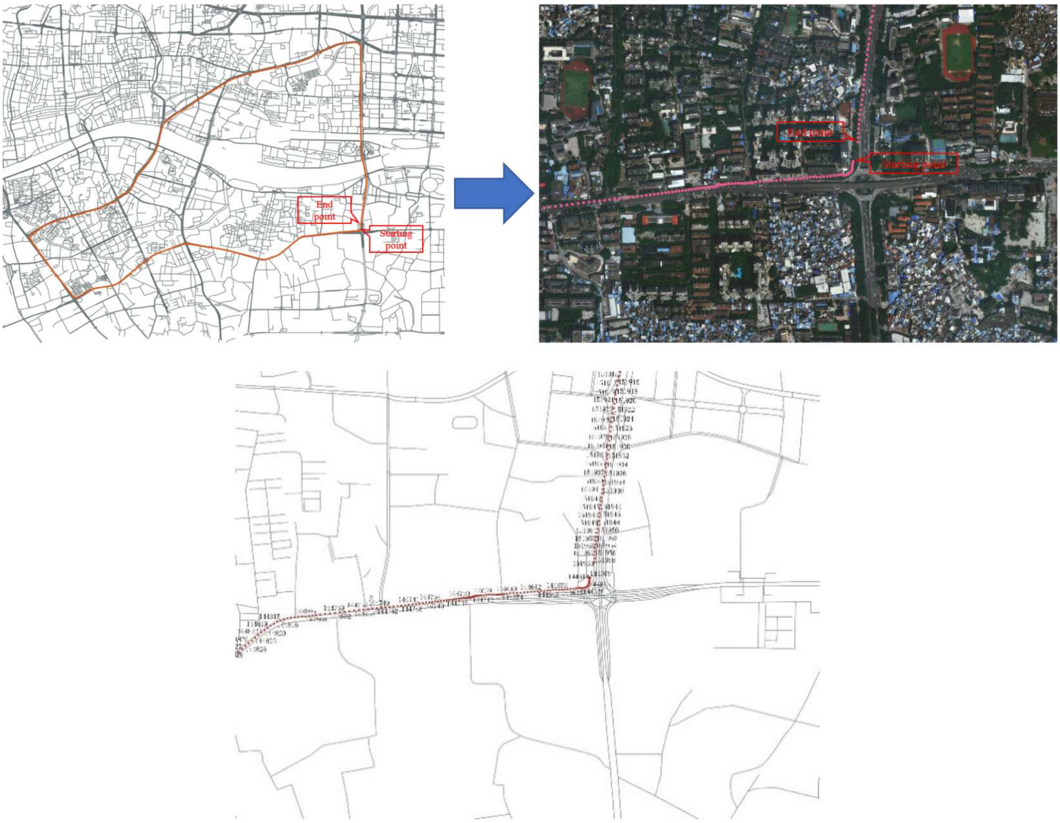

Figure 6. Dataset 1 GPS track route.

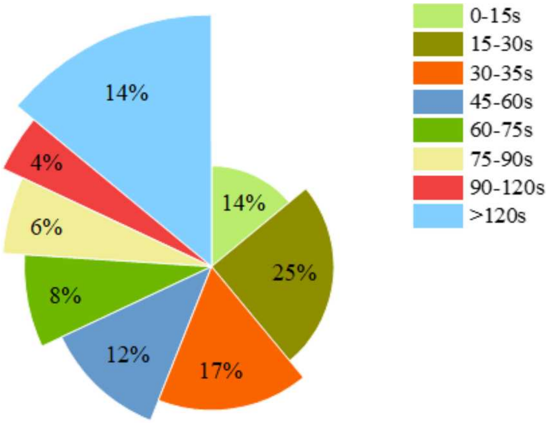

Figure 7. GPS data set sampling frequency distribution.

(2) The GPS open dataset of floating vehicles in the Dongguan demonstration area, the dataset collection area is located in the north-central region of Dongguan city, with the latitude and longitude range between 113.702–113.825° E and 22.994–23.994° N. The original GPS data contains several sampling frequencies (time intervals), from 10s to 360 s. However, most of them are concentrated within 45 s, and the sampling frequency distribution is shown in Figure 7.

The data structure of the GPS dataset trajectory points includes the moment of acquiring the information, longitude, latitude, the speed at which the floating vehicle is running and the number of the floating car. Some of the trajectory data of a particular floating car is shown in Table 1.

The city road network data used in the experiment is derived from the wiki map of the open-source project OSM. The city road network data within the corresponding latitude and longitude range are extracted according to the area where the GPS track is located. Some of the information extraction results are shown in Table 2.

In order to validate the accuracy of the hybrid model, the hybrid model was tested in the field as well as feasibility score. The trajectory data of a floating vehicle in the urban area of Ganzhou, China were collected using a GPS coordinate acquisition device, and the frequency of collecting the coordinates of the floating vehicle was between 5 and 10 s. The total length of the route traveled by the floating vehicle is about 17.3 km. The floating vehicle's GPS track data contains 752 track points.

3.2. Test environment settings

The algorithms utilized in the model were implemented on the Python 3.8 platform. The experimental results were visually displayed using the QGIS platform running on a Windows 10 64-bit operating system. The hardware setup included an Intel(R) Xeon(R) W-2123 CPU running

Table 1. Floating vehicle GPS track data set partial data.

Time	Longitude	Latitude	Speed(km/h)	Num
132702	113.304949	23.09824	30.86173	7
132703	113.304867	23.09823	30.38762	7
132704	113.304783	23.09822	31.13212	7
132705	113.304697	23.09821	31.88959	7
132706	113.304607	23.0982	32.73225	7

Table 2. Extracted feature information of some road network sections.

Road segment number	Longitude of the start of the road segment	Latitude of the start of the road segment	Longitude of the endpoint of the road segment	Latitude of the endpoint of the road segment	Length of a road segment (m)
624	113.2786092	23.09658003	113.283276	23.095694	488
770	113.2682924	23.10561768	113.2694865	23.10740711	234
373	113.2694865	23.10740711	113.2723276	23.105972	332
139	113.2743446	23.10777352	113.2692661	23.1072126	630
199	113.2713673	23.11022701	113.2692661	23.1072126	402

at 3.60 GHz and 32GB of memory. The model of the graphics card is NVIDIA Quadro P620. In this test environment, map matching was performed on dataset 1 and the time taken for matching was 0.423 s.

4. Results and discussion

4.1. Dilution results of trajectory data

The effect before and after dilution is shown in Figures 8 and 9. Figure 8(a) contains many residual points, redundant points generated by waiting for traffic lights at intersections. After dilution, as shown in Figure 8(b), it can better solve the problem of redundant invalid trajectory points. Figure 9 shows the result of trajectory dilution of a common road segment. The diluted trajectory can present the original trajectory's basic route to improve the model's operation efficiency with the guarantee of matching accuracy. Dataset 1 is compressed from 2031 trajectory points before compression to 479 trajectory points. Dataset 2 is not diluted due to uneven GPS and large sampling intervals in some periods.

4.2. Matching results for complex intersections and parallel road segments

After extraction and dilution processing, the vehicle driving trajectory data in dataset 1 are selected to demonstrate the processing results of the hybrid map-matching algorithm proposed in this paper. Based on the actual road width in dataset 1, the value of GPS positioning error σ_d in the observation probability of Equation (1) is 20. The original GPS trajectory matching results are shown in Figures 10 and 11, which represent the matching results of trajectory offset when the vehicle is driving at complex intersections and parallel road segments, respectively. In Figure 10,

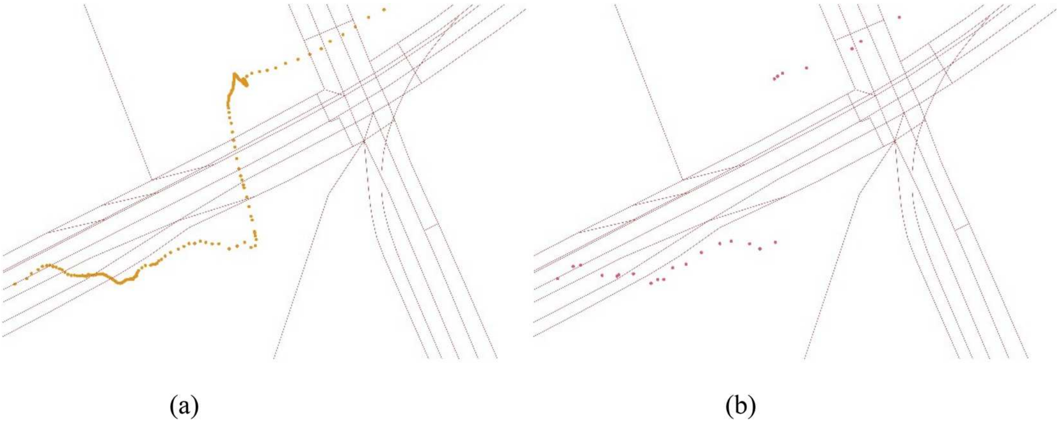

Figure 8. Scene 1 Comparison before and after track thinning.

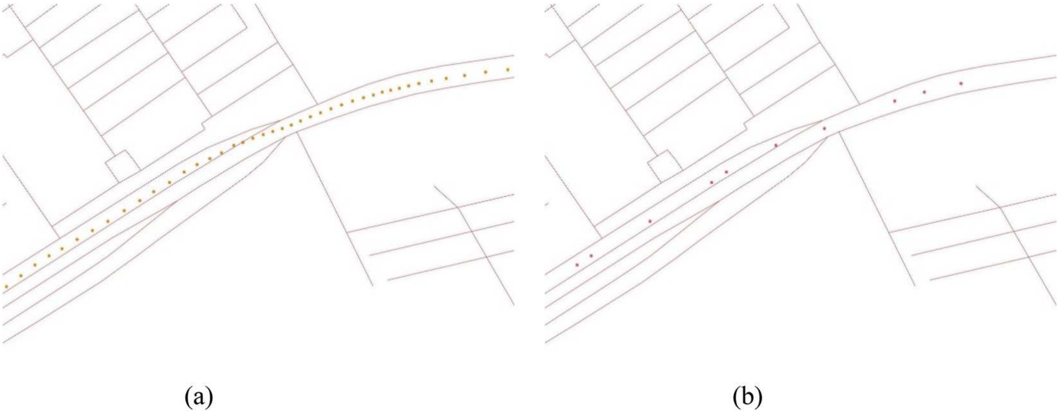

Figure 9. Scene 2 Comparison before and after track thinning.

the trajectory offset of the vehicle at the complex intersection is processed. The correct matching of the trajectory at the complex intersection is ensured because the vehicle driving direction and the angle between the vehicle direction and the direction vector of the candidate road segment are considered in the model based on the spatial positioning error. In Figure 11, the GPS trajectory is offset, i.e. it moves from one road to another in the parallel road, and at this time, if only the spatial positioning accuracy is considered, there will inevitably be wrong matching. In this paper, the shortest path distance of spatial accessibility is added as the limiting factor in the section transfer. Calculate the shortest path probability through Equation (5) and substitute it into Equation (6) to obtain the transition probability. The driver usually chooses the shortest path between the starting and ending points as the driving route, which ensures that the matched trajectory will not 'jump' back and forth on two parallel roads.

Figure 12 demonstrates the matching results of floating vehicle trajectories on complex road segments in the field test. The 'fork' in the figure indicates the trajectory of the floating vehicle before matching, and the yellow 'circle' indicates the trajectory of the floating vehicle after matching. The accuracy of the correct matching of the GPS sampling points is more than 97.2%. The time taken for the whole matching process in the test environment of section 3.2 is 0.509 s.

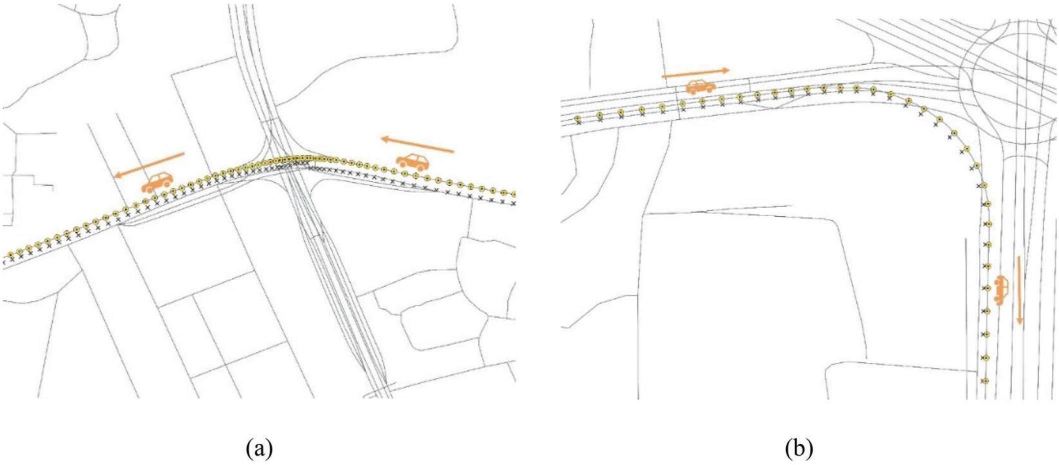

Figure 10. Matching results for complex intersection sections.

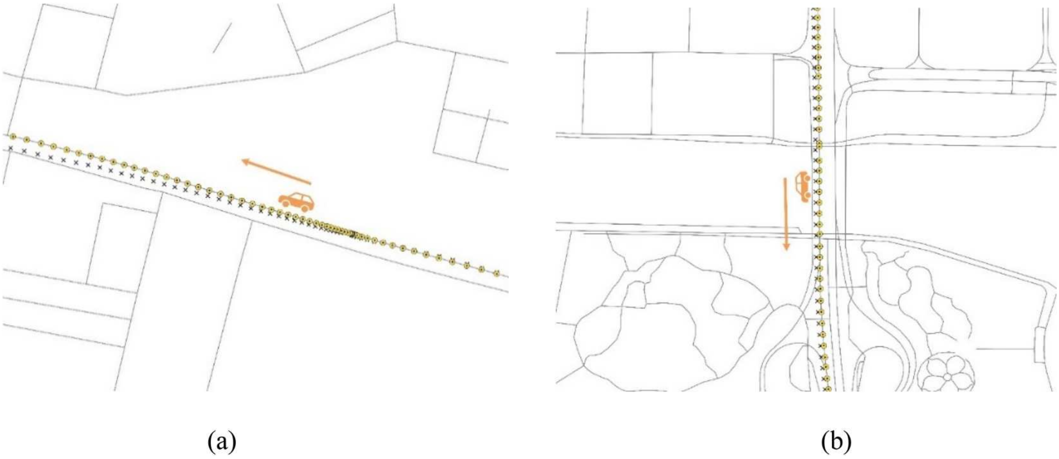

Figure 11. Matching results of parallel road sections.

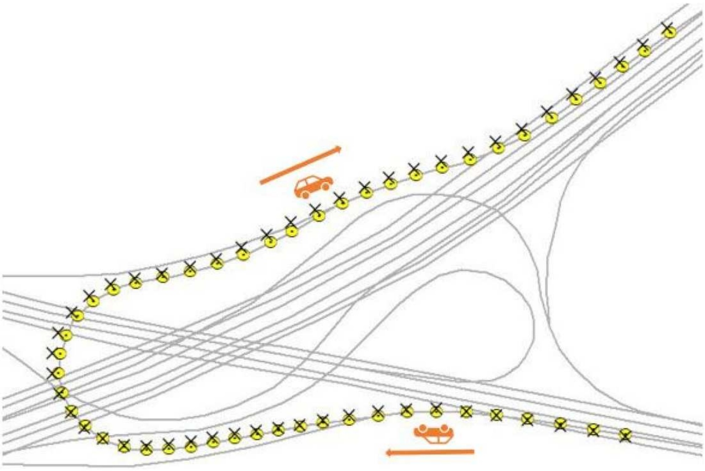

Figure 12. Matching results for complex road segments in field tests.

4.3. Matching results impact analysis

4.3.1. Effect of the number of candidate sections on matching results

Figure 13 shows the relationship between the number of candidate road segments and map matching quality. Because after increasing the number of candidate road segments, the model can find an optimal matching road segment from more candidate road segments. However, the matching accuracy will gradually stabilize when the candidate road segments reach a certain number. At this time, these candidate road segments must contain a real driving path. Although the traditional HMM model is better than the model proposed in this paper regarding matching time, it is necessary to sacrifice the matching efficiency of the model to obtain faster computation time. In offline map matching, the computation time of the model is not strictly pursued, but it is necessary to ensure that the model achieves good enough map matching quality (Tanaka et al. 2021).

As shown in Figure 14, the running time of the model proposed in this paper and the conventional HMM model gradually increases with the number of candidate road segments. The number of candidate road segments increases, and the average time taken by the algorithm to compute each

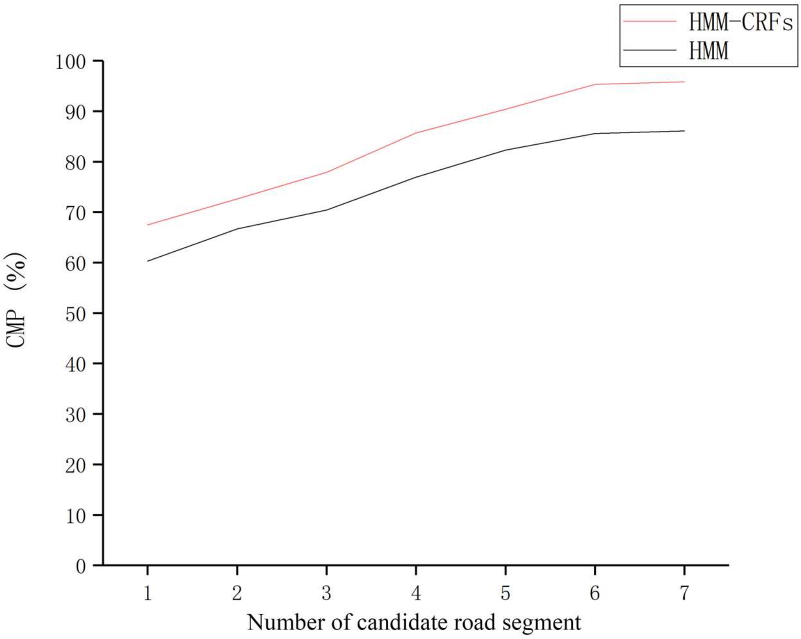

Figure 13. Comparison of the matching quality of different candidate sections and sampling points.

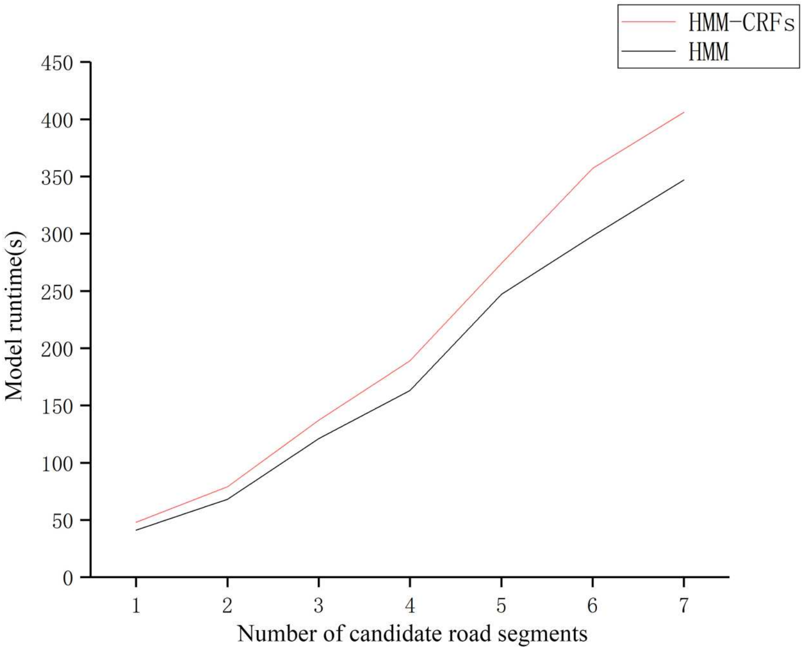

Figure 14. Comparison of model runtimes with different number of candidate roadway segments.

GPS track point also increases, resulting in a longer overall computation time. The figure shows that the HMM model outperforms the HMM-CRFs model proposed in this paper in terms of computation time. The main reason is that traditional HMM models only consider fewer constraints when

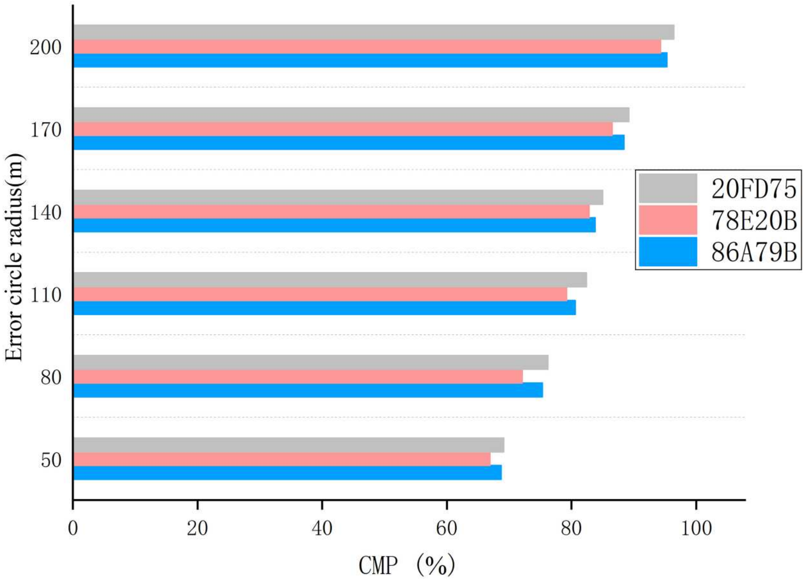

Figure 15. Matching accuracy with different error circle radii.

selecting spatiotemporal features of trajectories and road networks, and the computation time is shortened when calculating transfer probabilities and observation probabilities. However, despite the fact that the model in this paper employs trajectory compression and meshing algorithms to improve the performance of the model, it is not possible to improve the performance of the model. Forces the algorithm to run at reduced performance when computing multiple complex spatiotemporal features, but the computation time is also in an acceptable range for offline map matching.

4.3.2. Effect of candidate error circle setting on matching results

To further discuss the matching efficiency of the model under other conditions, the matching quality under different candidate error circle radii was set. The GPS trajectories of five cabs in Dataset 2 from 1 November 2015, to 10 November 2015, were selected, and six groups of error circle radii were set as 50, 80, 110, 140, 170, and 200, and the GPS trajectory matching accuracy of three cabs under each error circle radius were compared, as shown in Figure 15. As the radius of the error circle increases, the accuracy of GPS trajectory matching gradually improves, and in the model proposed in this paper, the accuracy of matching reaches the maximum when the radius of the error circle is set to 200 m. When setting the error circle radius threshold, the range needs to be reasonably determined, and the size of the value directly affects the calculation volume of the algorithm. When the radius R is set too large, it will make the calculation efficiency of the algorithm decrease and take more time to calculate all the candidate road segments within the error circle; while when the radius R is set too small, it will appear that the whole traveled road segments

Table 3. Comparison of matching accuracy of different algorithms.

Algorithm	Matching accuracy		
	80 m	140 m	200 m
HMM	66.9%	73.5%	88.2%
EHMM	69.7%	76.8%	90.3%
HMM-CRFs	76.2%	85.1%	96.4%
ST-Matching	71.1%	81.3%	92.6%

may not fall into the candidate circle area, and the matched road segments are wrong sections at this time. When the radius R is set too small, the real traveled road may not fall into the candidate circle area, and the matched road is wrong.

4.4. Matching accuracy comparison

To validate the performance and matching accuracy of the map-matching algorithm proposed in this paper, it was compared with other similar algorithms based on the HMM model. Two algorithms were considered for comparison: The first one is an ST-Matching algorithm that combines temporal and spatial features, incorporating geometric topological constraints and speed limits. The second algorithm introduces traffic rules and utilizes historical Floating Car Data (FCD) information, called the enhanced Hidden Markov Map Matching (EHMM) model. In this study, the hybrid map matching algorithm developed was compared with the traditional HMM, EHMM, and ST-Matching algorithms using Dataset 1 under the same conditions. Three error circle radii, namely 80, 140, and 200 m, were used to evaluate the matching accuracy of these algorithms. The results presented in Table 3 demonstrate that under the condition of using the same dataset, the hybrid HMM-CRFs map matching method improves the matching accuracy by 8.2%, 6.1%, and 3.8% compared to the traditional HMM, EHMM algorithm, and ST-Matching algorithm, respectively, when the radius of the error circle is 200 m. Consequently, the algorithm introduced in this paper outperforms the other three algorithms in terms of matching accuracy.

5. Conclusions

For the existing map-matching algorithms, the HMM has shown satisfactory results in terms of map-matching quality and matching rate, even when dealing with low-frequency GPS sampling data. However, the inherent ‘annotation bias’ problem of the HMM makes some erroneous results when dealing with the inter-segment transfer. To address this issue, this paper introduces Conditional Random Fields (CRFs) based on the HMM model. By incorporating global spatial–temporal context information and recalculating the weights, observation probability, and transfer probability of the HMM, better matching performance is achieved. The experimental results show that the accuracy of the method for matching at an error circle radius of 200 m is 96.4%. Furthermore, the average computation time for each candidate road segment remains within 50 ms. Additionally, the study analyzes the impact of different numbers of candidate road segments and error circle radii on the matching accuracy using Dataset2. The results indicate that this method outperforms simple geometric map-matching and directionality-based approaches, exhibiting strong robustness in handling GPS track points on complex intersections and parallel roads. Compared to traditional HMM, EHMM, and ST-Matching methods, this approach considers the relationship between map matching accuracy and efficiency, achieving improved matching rates while maintaining high accuracy. Future research will focus on enhancing the map-matching accuracy at lower sampling frequencies and ensuring real-time performance.

Code availability section

Name of the code: HMM-CRFs hybrid model
Hardware requirements: NA
Program language: Python
Software required: NA
Program size: 316 kb
The source codes are available at: https://github.com/giserLhc/MapMatching_baseCRFS.git

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the National Natural Science Foundation of China [grant number 42261072].

Data availability statement

The data that support the findings of this study are publicly available at (https://www.openits.cn/openData2/570.jhtml) for GPS track dataset and (https://www.openits.cn/openData2/604.jhtml) for GPS open dataset of floating vehicles.

ORCID

Youliang Chen http://orcid.org/0000-0002-0319-7051

References

Alejandra Blazquez, Carola, Jana Ries, Roberto Jesus Leon, and Pablo Andres Miranda. 2018. “An Instance-Specific Parameter Tuning Approach Using Fuzzy Logic for a Post Processing Topological Map-matching Algorithm.” *IEEE Intelligent Transportation Systems Magazine* 10 (4): 87–97. https://doi.org/10.1109/MITS.2018.2867527.

Berger, Adam. 1998. “The Improved Iterative Scaling Algorithm: A Gentle Introduction.”

Bierlaire, Michel, Jingmin Chen, and Jeffrey Newman. 2013. “A Probabilistic Map Matching Method for Smartphone GPS Data.” *Transportation Research Part C-Emerging Technologies* 26: 78–98. https://doi.org/10.1016/j.trc.2012.08.001.

Blazquez, Carola A., and Alan P. Vonderohe. 2005. “Simple Map-matching Algorithm Applied to Intelligent Winter Maintenance Vehicle Data.” *Transportation Research Record* 1935 (1): 68–76. https://doi.org/10.1177/0361198105193500108.

Bloit, Julien, and Xavier Rodet. 2008. *Short-time Viterbi for Online HMM Decoding: Evaluation on a Real-time Phone Recognition Task.*

Che, Mingliang, Yingli Wang, Chi Zhang, and Xinliang Cao. 2018. “An Enhanced Hidden Markov Map Matching Model for Floating Car Data.” *Sensors* 18 (6): 1758. https://doi.org/10.3390/s18061758.

Harrison, F., T. Burgoine, K. Corder, E. M. van Sluijs, and A. Jones. 2014. “How well do Modelled Routes to School Record the Environments Children Are Exposed to?: A Cross-sectional Comparison of GIS-modelled and GPS-measured routes to school.” *International Journal of Health Geographics* 13 (1): 5. https://doi.org/10.1186/1476-072X-13-5.

Hashemi, Mahdi. 2017. “Reusability of the Output of Map-matching Algorithms across Space and Time Through machine Learning.” *Ieee Transactions on Intelligent Transportation Systems* 18 (11): 3017–3026. https://doi.org/10.1109/TITS.2017.2669085.

Huang, Jian, Jinhui Qie, Chunwei Liu, Siyang Li, Jingnong Weng, and Weifeng Lv. 2015. “Cloud Computing-based Map-matching for Transportation Data Center.” *Electronic Commerce Research and Applications* 14 (6): 431–443. https://doi.org/10.1016/j.elerap.2015.03.006.

Jagadeesh, George R., and Thambipillai Srikanthan. 2017. “Online Map-matching of Noisy and Sparse Location Data with Hidden Markov and Route Choice Models.” *IEEE Transactions on Intelligent Transportation Systems* 18 (9): 2423–2434. https://doi.org/10.1109/TITS.2017.2647967

Jonietz, David, Vyron Antonio, Linda See, and Alexander Zipf. 2017. “Highlighting Current Trends in Volunteered Geographic Information.” *ISPRS International Journal of Geo-Information* 6 (7): 202. https://doi.org/10.3390/ijgi6070202.

Jung, Jin-Woo, Byung-Chul So, Jin-Gu Kang, Dong-Woo Lim, and Yunsik Son. 2019. “Expanded Douglas-Peucker Polygonal Approximation and Opposite Angle-Based Exact Cell Decomposition for Path Planning with Curvilinear Obstacles.” *Applied Sciences-Basel* 9 (4): 638. https://doi.org/10.3390/app9040638.

Karamete, Bilge Kaan, Louai Adhami, and Eli Glaser. 2021. “An Adaptive Markov Chain Algorithm Applied Over Map-matching of Vehicle Trip GPS Data.” *Geo-Spatial Information Science* 24 (3): 484–497. https://doi.org/10.1080/10095020.2020.1866956.

Li, Dengao, Xuan Jia, and Jumin Zhao. 2020. “A Novel Hybrid Fusion Algorithm for Low-cost GPS/INS Integrated Navigation System During GPS Outages.” *IEEE Access* 8:53984–53996. https://doi.org/10.1109/ACCESS.2020.2981015.

Luo, An, Shenghua Chen, and Bin Xv. 2017. “Enhanced Map-matching Algorithm with a Hidden Markov Model for Mobile Phone Positioning.” *Isprs International Journal of Geo-Information* 6 (11): 327. https://doi.org/10.3390/ijgi6110327.

Nassreddine, Ghalia, Fahed Abdallah, and Thierry Denoeux. 2008. *Map Matching Algorithm using belief function theory: HEUDIASYC, Université de Technologie de Compiègne, Compiègne*. ÃŽle-de-France, FR: FRANCE Universite de Technologie de Compiegne, Compiegne.

Newson, Paul, and John Krumm. 2009. *Hidden Markov Map Matching through Noise and Sparseness*. Redmond, WA: Microsoft Corporation.

Pereira, Francisco Camara, Hugo Costa, and Nuno Martinho Pereira. 2009. “An Off-line Map-matching Algorithm for Incomplete Map Databases.” *European Transport Research Review* 1 (3): 107–124. https://doi.org/10.1007/s12544-009-0013-6.

Preiss, James A., Karol Hausman, Gaurav S. Sukhatme, and Stephan Weiss. 2018. “Simultaneous Self-calibration and Navigation Using Trajectory Optimization.” *International Journal of Robotics Research* 37 (13–14): 1573–1594. https://doi.org/10.1177/0278364918781734.

Quddus, Mohammed A., Washington Y. Ochieng, and Robert B. Nol. 2007. “Current Map-matching Algorithms for Transport Applications: State-of-the art and Future Research Directions.” *Transportation Research Part C: Emerging Technologies* 15 (5): 312–328. https://doi.org/10.1016/j.trc.2007.05.002.

Sama, Kyle, Yoichi Morales, Naoki Akai, Eijiro Takeuchi, and Kazuya Takeda. 2018. *Learning How to Drive in Blind Intersections from Human Data: Institute of Innovation for Future Society*. Nagoya: Nagoya University, Japan Graduate School of Information Science, Japan Takeda Lab.

Santhosh, K. K., D. P. Dogra, and P. P. Roy. 2021. “Anomaly Detection in Road Traffic Using Visual Surveillance: A Survey.” *Acm Computing Surveys* 53 (6): 1–26. https://doi.org/10.1145/3417989.

Sarvrood, Yashar Balazadegan, Fei Liu, and Yang Gao. 2018. “Tight Integration of Kinematic Precise Point Positioning and Digital Map for Land Vehicle Localisation.” *Survey Review* 50 (362): 416–424. https://doi.org/10.1080/00396265.2017.1303019.

Shen, Zhihao, Wan Du, and Xi Zhao. 2020. *DMM: Fast Map Matching for Cellular Data*.

Shen, Li, and Peter R. Stopher. 2014. “Review of GPS Travel Survey and GPS Data-Processing Methods.” *Transport Reviews* 34 (3): 316–334. https://doi.org/10.1080/01441647.2014.903530.

Singh, Saravjeet, Jaiteg Singh, and Sukhjit Singh Sehra. 2020. “Genetic-Inspired Map Matching Algorithm for Real-Time GPS Trajectories.” *Arabian Journal for Science & Engineering (Springer Science & Business Media B.V.)* 45 (4): 2587–2603. https://doi.org/10.1007/s13369-019-04247-1.

Singh, Jaiteg, Saravjeet Singh, Sukhjit Singh, and Hardeep Singh. 2019. “Evaluating the performance of map matching algorithms for navigation systems: an empirical study.” *Spatial Information Research* 27 (1): 63–74. https://doi.org/10.1007/s41324-018-0214-y.

Tanaka, Akira, Nariaki Tateiwa, Nozomi Hata, Akihiro Yoshida, Takashi Wakamatsu, Shota Osafune, and Katsuki Fujisawa. 2021. “Offline Map Matching Using Time-expanded Graph for Low-frequency Data.” *Transportation Research Part C-Emerging Technologies* 130:103265. https://doi.org/10.1016/j.trc.2021.103265.

Wang, Jinyue, Martin Metzner, and Volker Schwieger. 2020. “Potential Enhancement for Wrong-way Driver Detection using Precise Attribute Information.” *Journal of Navigation* 73 (2): 384–397. https://doi.org/10.1017/S0373463319000663.

Wang, Hongjian, Xianfeng Tang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. 2019. “A Simple Baseline for Travel Time Estimation using Large-scale Trip Data.” *Acm Transactions on Intelligent Systems and Technology* 10 (2), https://doi.org/10.1145/3293317.

Wei, Hong, Yin Wang, George Forman, and Yanmin Zhu. 2013. “Map matching: comparison of approaches using sparse and noisy data." SIGSPATIAL’13: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Orlando Florida, USA.

White, C. E.*, D. Bernstein, and A. L. Kornhauser. 2000. “Some Map Matching Algorithms for Personal Navigation Assistants.” *Transportation Research. Part C: Emerging Technologies* 8 (1–6): 91–108. https://doi.org/10.1016/S0968-090X(00)00026-7.

Xu, Chunhui, Anqin Zhang, and Yu Chen. 2020. “Traffic Congestion Forecasting in Shanghai Based on Multi-period Hotspot Clustering.” *IEEE Access* 8:63255–63269. https://doi.org/10.1109/ACCESS.2020.2983184.

Zhang, Haiyan, Yonglong Luo, Qingying Yu, Xiaoyao Zheng, and Xuejing Li. 2021. “Map-matching approach based on link factor and hidden Markov model.” *Journal of Intelligent & Fuzzy Systems* 40 (3): 5455–5471. https://doi.org/10.3233/JIFS-202292.

Zhao, Xiangmo, Xin Cheng, Jingmei Zhou, Zhigang Xu, Nilanjan Dey, Amira S. Ashour, and Suresh Chandra Satapathy. 2018. “Advanced Topological Map Matching Algorithm Based on D-S Theory.” *Arabian Journal for Science and Engineering* 43 (8): 3863–3874. https://doi.org/10.1007/s13369-017-2569-0.

Zhao, Kai, Jie Feng, Zhao Xu, Tong Xia, Lin Chen, Funing Sun, Diansheng Guo, Depeng Jin, and Yong Li. 2019. “DeepMM: Deep Learning Based Map Matching with Data Augmentation." SIGSPATIAL ’19: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Chicago IL USA.

Zheng, K., Y. Zheng, X. Xie, X. F. Zhou, and IEEE. 2012. “Reducing Uncertainty of Low-Sampling-Rate Trajectories.” *2012 IEEE 28th International Conference on Data Engineering (ICDE)*, 1144–1155. https://doi.org/10.1109/ICDE.2012.42.