HAUM^3: A Height Aware Urban Map Matching Mechanism

Jie Tang[1], Sunjian Zheng[1(✉)], Bo Yu[2], and Shaoshan Liu[2]

[1] South China University of Technology, Guangzhou, China
cstangjie@scut.edu.cn, sunjian_zheng@163.com
[2] PerceptIn, Fremont, USA
{bo.yu,shaoshan.liu}@perceptin.io

Abstract. Map matching is an essential component of urban intelligent transportation, providing fundamental data for technologies such as path planning, traffic analysis, and trajectory analysis. However, our commercial deployments showed that existing map matching algorithms behave ineffectively in complex urban scenarios such as elevated and interchange roads due to their insensitivity to the road height change. To improve the map matching for urban map services, we proposed and deployed **HAUM^3**, a novel Height Aware Urban Map Matching Mechanism with Spatio-Temporal Correlation. Firstly, starting from the dynamic driving process of vehicles, an urban driving scene classification model is designed based on the images and sensor data generated synchronously by the vehicles. This model correlates the spatio-temporal state changes of vehicles on roads of different heights. After that, according to the characteristics of different types of map matching algorithms, the classification model is fused into traditional map matching algorithms, enabling them to aware roads at different heights and improving their performance. Evaluation results show that this mechanism significantly improves the accuracy of different types of map matching algorithms in complex urban roads, with an average accuracy increase of over 10%, enhancing the usability of traditional map matching algorithms.

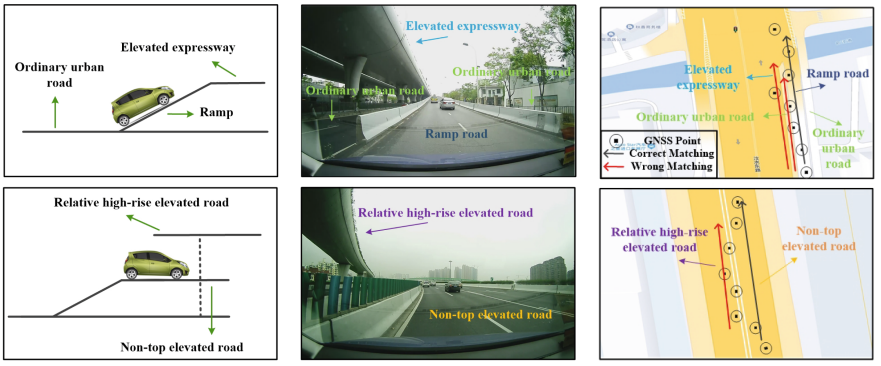
Keywords: Map matching · Driving scene classification · Images and sensor data · Height awareness

1 Introduction

Map matching is a technique that associates a series of GNSS (Global Navigation Satellite System) points with a road network by using maps to constrain GNSS data. GNSS sensors have been widely deployed in cars, smartphones, and other mobile devices to collect movement information. These devices accumulate vast amounts of trajectory data, which can be pre-processed using map matching technology to assign semantic information about the road network to raw data. This can then be applied in various fields such as managing urban traffic congestion, identifying functional areas in cities, and urban route planning [1].

A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 593, pp. 505–525, 2024.
https://doi.org/10.1007/978-3-031-63989-0_27

However, with the rapid increase in automobiles, the traffic infrastructure has been fully developed and then urban road networks get complicated dramatically. Cities worldwide have constructed many elevated and interchange roads to alleviate the enormous traffic pressure on the roads. For example, Shanghai has more than 622 km of elevated roads, including multi-level elevated roads. As a result, in a two-dimensional map application, elevated roads may overlap with ground-level roads. The accuracy of position and movement direction provided by GNSS is even lower under the influence of elevated roads, with a maximum error of over 15 m [2]. Therefore, it is difficult to determine a vehicle's trajectory when traveling on elevated and nearby roads. That leaves a big problem for urban map matching both in accuracy and robustness.



(a) Side cut of the vehicle position (b) Images of the road from the driving viewpoint (c) Visualization of matching results

Fig. 1. Map mis-matching cases due to height difference. Subplot (c) shows the matching results of vehicle trajectories in complex urban roads obtained by applying the map matching algorithm. The overlap of roads at different heights and the noisy GNSS sampling points will lead to different degrees of mis-matching: the upper part of subplot (c) indicates that the matching algorithm mis-matches a normal urban road and a elevated expressway; the lower part of subplot (c) indicates that the matching algorithm mis-matches a relative high-rise elevated road.

When we dive into deployment details, we can find that in complex urban road scenarios, erroneous map mismatches of traditional map matching methods based on GNSS sampling points and road network structures can be blamed on their blindness to the road height differences. According to our experiments from commercial deployment, mismatches often occur on elevated and nearby roads, and Fig. 1 demonstrates the typical map mis-matching cases. When the vehicle is driving on the ramp road, the GNSS trajectory may be shifted to the nearby ordinary urban road as well as the elevated road, which leads to the map matching algorithm mis-matching the ordinary urban road and the elevated road, as shown in the upper part of Fig. 1; when the vehicle is driving on the elevated expressway, the GNSS trajectory may be shifted to the nearby elevated road on the upper or lower level, as shown in the lower part of Fig. 1, which demonstrates the map matching algorithm mis-matching the elevated road on

the upper level. Even though, currently, there have some methods for detecting elevated roads, which are based on altitude [3] or features [4,5]. However, they are challenged in the accuracy or robustness, therefore, they cannot effectively work for urban roads with heights difference.

The erroneous map matching problems in complex urban roads generate great need for height information in the GNSS guided map matching. With that observation, based on a large amount of deployment data, we have developed and deployed **HAUM^3**, a novel Height Aware Urban Map Matching Mechanism with Spatio-Temporal Correlation. By combining the dynamic running process of vehicles with spatio-temporal images and sensor data generated by vehicles, it can obtain the information of road types at different heights and incorporate them into existing map matching methods. It therefore enables map matching algorithms to distinguish roads of different heights effectively and can improve the matching performance of existing map matching methods on complex urban roads. The main contributions of our work can be concluded as follows:

1. A map matching mechanism named **HAUM^3** has been proposed. This mechanism utilizes the images and sensor data of vehicles, combined with the spatio-temporal state of the vehicle, to overcome the issue of low sensitivity in map matching algorithms for complex urban roads. It effectively enhances the matching effectiveness of elevated areas, such as expressways, ramps, and auxiliary roads.
2. A spatio-temporal correlated urban road scene classification model has been devised by integrating real-time in-vehicle images and sensor data. This model amalgamates the temporal and spatial attributes of the dynamic vehicle operation process and achieves effective discrimination among five distinct levels of urban roads: high-rise elevated roads, non-top elevated roads, ramps, ordinary urban roads, and tunnels, through the simultaneous utilization of vehicle-captured images and sensor data.
3. A map matching algorithm with superior performance can be obtained. By deploying **HAUM^3** on different types of map matching algorithms, they all can obtain some performance improvement. For global map matching algorithms, stronger robustness can be obtained, while for local/incremental map matching algorithms, faster convergence can be achieved.

2 Related Work

2.1 Map Matching

Scholars have proposed various map matching algorithms and classification systems for map matching algorithms to address different characteristics of GNSS data and map data. Based on the range of sample points considered when matching trajectories, map matching algorithms can be divided into local/incremental and global methods [6].

Local/incremental algorithms use the local features of the current trajectory point to match the road network, using a greedy strategy to gradually move

towards the global solution from the existing matching solution. For example, Brakatsoulas et al. [7] proposed a map matching algorithm based on the similarity and turning angle between trajectory segments and candidate paths. Fu et al. [8] improved the map matching algorithm using the extended Viterbi algorithm and the adaptive sliding window mechanism, considering drivers' travel preferences, road class, network topology, and other factors. Local/incremental algorithms are fast in calculation speed and strong in real-time performance, so they are often used in applications with online demands.

Global algorithms consider the entire sampling trajectory and find the closest matching path from the road network. A classic method is the map matching algorithm based on the Hidden Markov Model (HMM) [9], transforming the map matching problem into a decoding problem. Gui et al. [10] proposed a trajectory segmentation Hidden Markov Model map matching method based on heading homogeneity. Compared with local/incremental algorithms, global algorithms are more accurate, show greater robustness to reduced sampling rates, and can better handle map matching problems with long sampling intervals. Nevertheless, they have higher computational costs and are only applicable for offline matching [11].

2.2 Detection of Elevated Roads

As cities continue to develop and electronic map technology advances, urban electronic maps are becoming increasingly complex, especially on trunk roads and elevated expressways where roads at different heights overlap and interweave on a 2D map. The number of candidate road segments that can be found from a single sampling point increases rapidly. Therefore, the problem of elevated road matching has attracted wide attention from scholars. Ho et al. [3] used an altitude-based method to identify the position and status of vehicles, but using the elevation directly for position detection is not feasible due to the very low accuracy of barometric pressure sensors. In contrast, Gong et al. [4] proposed a feature-based method better to capture the differences between elevated and ground roads. They extract features, such as variance in vehicle speed and the number of tracked satellites, and input them into a classification model to identify whether a vehicle is on an elevated road. Based on this, Zhang et al. [5] proposed an elevated road network, which learns two feature descriptors and uses four features (satellite plane projection feature, group ID, distance of elevated road, and sequence speed) to distinguish between surface roads and elevated roads. Furthermore, Fu et al. [12] use image data from driving recorders to classify driving scenes and assist map matching in complex urban road environments.

None of the map matching algorithms based on traditional methods take into account the adverse effects caused by urban roads of different heights. Methods specifically designed for detecting elevated roads have drawbacks such as low accuracy and poor robustness and cannot effectively distinguish between the different heights of urban roads. The mechanism proposed in this paper combines images and sensor information from the dynamic driving process of a

vehicle with spatio-temporal state, endowing the map matching algorithm with height awareness, thus improving the accuracy of the map matching algorithm in complex urban road environments.

3 Problem Definition and Mechanism Overview

In order to address the issue of mis-matching due to height differences of map matching algorithms in complex urban roads, this paper proposes utilizing coherent image and sensor information to extract the vehicle's height variation during its journey. The data are derived from vehicle devices with travel records or driverless cars. During the driving process, the vehicle is continuously positioned according to a particular sampling frequency while corresponding video frames and sensor data are recorded. By combining the trajectory variations of the vehicle on different height roads and the images and sensor information generated during the process, the perception and differentiation of roads at different heights are achieved, ultimately obtaining a height aware map matching algorithm. The relevant definitions are as follows:

Definition 1: Urban Road Network. $G = (V, E)$ represents a directed graph of the urban road network. The vertices V denotes intersections or endpoints in the road network, while the edges E correspond to directed road segments. Each segment has its corresponding road attribute information.

Definition 2: Sampling Points. $t = (timestamp, lon, lat, image, sensor\ data)$ represents a set of data collected by a single device using its positioning module to obtain the current location information while synchronously recording the video and sensor data of the current moment. The *timestamap* denotes the time at which the sampling point is taken, and (lon, lat) denotes the geographic location of the traveling device at that moment. The *image* and *sensor data* refer to the images and other vehicle-related sensor data synchronized with the sampling point collection.

Definition 3: Vehicle trajectory. $T = \{t_1, t_2, t_3, \ldots, t_n\}$ represents a set of samples obtained by the exact vehicle at a specific time interval in a period, where n denotes the number of sampling points.

Definition 4: Actual Path. $P = [e_1, e_2, \ldots, e_n]$ represents a sequence of actual paths that a vehicle traverses during its motion, where $\forall e_n \in E$ and n denotes the number of actual paths.

Definition 5: Distinction of Different Height Roads. $R = [r_1, r_2, r_3, \ldots, r_n] = S[t_1.(i,s), t_2.(i,s), \ldots, t_n.(i,s)]$ represents the road type results obtained from processing images and sensor data collected at different times and heights. Here, r_k denotes the road type result calculated at the kth sampling point, S is the processing mechanism, and $t_k.(i,s)$ denotes the image and sensor data obtained at the kth sampling point, where $k \epsilon [1, n]$.

Definition 6: Matching Path. $P' = M(G, T, R) = [e_1', e_2', \ldots, e_m']$ represents the matching path obtained by mapping vehicle trajectory T onto the urban road network G through the height aware map matching algorithm M, combining the results of distinction of different height roads R, where $\forall e_m' \in E$ and m denotes the number of matched road sections. The height aware map matching algorithm M aims to obtain a matching path P' closer to the actual trajectory P.

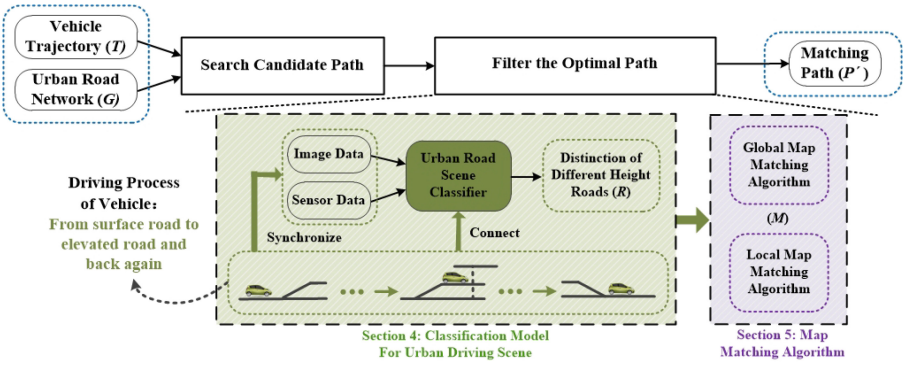


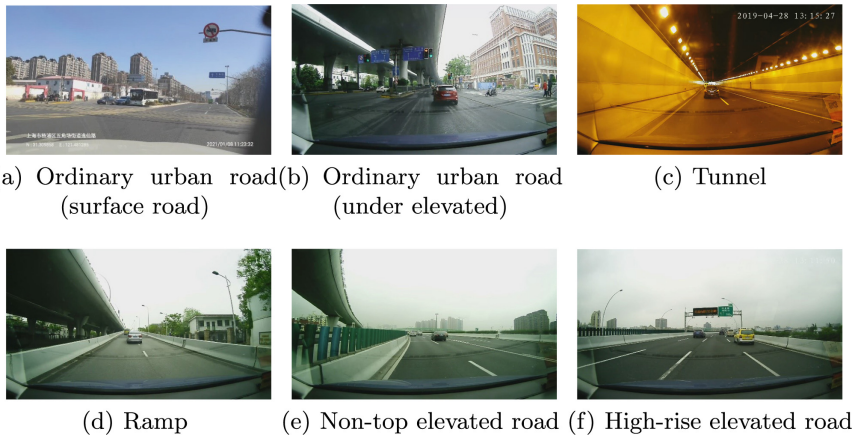
Fig. 2. The overall framework of the **HAUM³**. The upper part shows the overall operation of the map matching algorithm with height awareness; the lower part shows the schematic diagram of filtering the optimal path. Firstly, the urban road scene classifier model that links the dynamic driving process of vehicles obtains the distinction of different height roads. Then the different types of map matching algorithms are fused with the distinction of different height roads, endowing the map matching algorithm with height awareness. The optimal path is finally filtered.

The above definitions lead to **HAUM³**, a Height Aware Urban Map Matching Mechanism with Spatio-Temporal Correlation we have developed and deployed in the past year, and the overall framework is shown in Fig. 2. The map matching algorithm mainly consists of two steps. First, it searches for candidate paths in the map using the input trajectory information. After that, it selects the optimal matching path based on the rules and algorithms, and finally obtains the matching path. The proposed mechanism fuses the spatio-temporal correlation urban road scene classifier into searching for the optimal path, so that the optimal path searched is the one that matches the current actual road type. The classifier associates the vehicle's spatio-temporal state with the urban road scenes from the perspective of the vehicle's dynamic driving process. During the driving process, the images and sensor information collected synchronously by the vehicle are inputted into the urban road scene classifier. The obtained results are fused with the map matching algorithm to endow the map matching algorithm with height awareness, which leads to a better optimal matching path. The details are described in the following Sects. 4 and 5.

4 Classification Model for Urban Road Driving Scene

4.1 Scene Classification and Inspiration

The driving scene refers to the road environment in which the vehicle is moving. Due to current deficiencies in the map matching algorithm in complex urban roads, this paper first divides the vehicle driving scene into three main categories: elevated roads (including elevated expressways and ramps), ordinary urban roads (including elevated auxiliary roads and ordinary road scenes near elevated roads), and indoor roads (including tunnels and indoor parking area). In the elevated road scenes, they are further divided into ramp roads and elevated expressways. The elevated expressway scenes are further divided into relatively high-rise elevated roads (relatively high-rise interchanges) and non-top elevated roads (non-top interchanges). Figure 3 shows the images corresponding to the above five driving scenes.



(a) Ordinary urban road (surface road) (b) Ordinary urban road (under elevated) (c) Tunnel

(d) Ramp (e) Non-top elevated road (f) High-rise elevated road

Fig. 3. Various driving scenes recorded images.

Based on the above classification, the following urban driving scene indicator matrix is defined:

$$\boldsymbol{P}_{i,rtype} = [P_{i,ordinary} \quad P_{i,indoor} \quad P_{i,ramp} \quad P_{i,high} \quad P_{i,low}]$$

In the above equation, i denotes the ith sampling point, and $P_{i,ordinary}$, $P_{i,indoor}$, $P_{i,ramp}$, $P_{i,high}$, $P_{i,low}$ denotes the probability that the current sampling point belongs to the ordinary urban road, indoor road, ramp road, relative high-rise elevated road and non-top elevated road, respectively.

Note that in the following, O, I, R, H, and L denote the above five road types, respectively.

Through a large number of observations, this paper finds that vehicles driving in complex urban road scenarios satisfy the following laws: vehicles can reach any elevated road from ordinary urban roads through ramp roads, and then from elevated roads through ramp roads can go up or down to elevated roads, or return to ordinary urban roads. The specific diagram is shown in Fig. 4.

4.2 Urban Road Scene Classification Model with Spatio-Temporal Correlation

Inspired by the abovementioned laws, this paper proposes an urban road scene classification model based on images and sensor information with spatio-temporal correlation. The model inputs time-series images and sensor data, producing a matrix of driving scene indicators for each moment. Given the hierarchical relationship among different driving scenes, the model generates three sub-classifiers based on the granularity of the scenes:

The sub-classifier C_1 takes in image information and outputs the probabilities of three types of road scenes: elevated road, ordinary road, and indoor road, which are defined as P^{1_1}, P^{1_2}, and P^{1_3}, respectively. Specifically, $P_{i,normal} = P_i^{1_2}$ and $P_{i,indoor} = P_i^{1_3}$.

The sub-classifier C_2 confines the driving scene to elevated road, taking in sensor information such as vehicle speed and pitch angle (roll angle/lateral angle) information. It then outputs to assist in determining the probability of the following two road scenes: ramp road as well as elevated expressway, defined as P^{2_1} and P^{2_2}. Specifically, $P_{i,ramp} = P_i^{1_1} \cdot P_i^{2_1}$.

The sub-classifier C_3 confines the driving scene to elevated expressway, taking in image information, and outputs to assist in determining the probability of the following two road scenes: relative high-rise elevated road and non-top elevated road, defined as P^{3_1} and P^{3_2}. Specifically, $P_{i,high} = P_i^{1_1} \cdot P_i^{2_2} \cdot P_i^{3_1}$, and $P_{i,low} = P_i^{1_1} \cdot P_i^{2_2} \cdot P_i^{3_2}$.

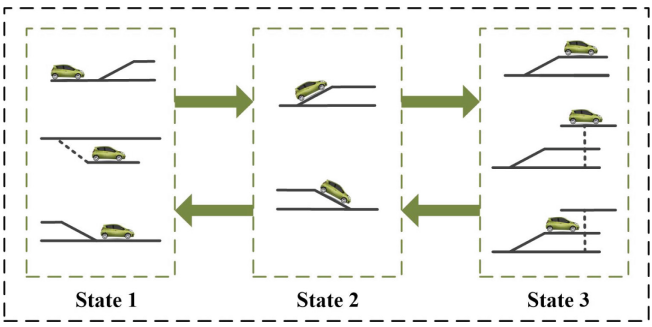


Fig. 4. Schematic diagram of the switching of vehicle driving states in different space-time states. The scenes of state 1 include: ordinary urban roads and indoor roads; the scenes of state 2 include: ramp roads; the scenes of state 3 include: relatively high-rise elevated roads and non-top elevated roads. The combination of scenes in various states can portray all the vehicle dynamic driving process.

The calculation process of the three sub-classifiers derived from the urban driving scene classification model based on images and sensor information with spatial-temporal correlation is shown in Fig. 5.

When the vehicle travels on the urban road network or the current map matching algorithm finds multiple matching roads, the mechanism process is initiated, and images and sensor data are collected synchronously. Firstly, the

collected image data is input to classifier C_1. Suppose the computed matching result is an ordinary urban or indoor road, corresponding to state 1. In that case, the probability result can be directly output, and the subsequent process is not continued. If the computed matching result is an elevated road $\left(P^{1_1} > V_1,\ simultaneously\ activating\ classifier\ C_2\right)$ and combined with the output of classifier C_2, it can be judged whether the vehicle is on a ramp section $\left(P^{1_1} \cdot P^{2_1} > V_2\right)$, corresponding to state 2. The probability result is output, and the process continues to run. Assuming that the vehicle has passed through the ramp section and is driving on the road above the elevated road, combined with the output of classifier C_2, it can be determined that the vehicle is currently on the elevated expressway $\left(P^{1_1} \cdot P^{2_2} > V_3,\ simultaneously\ activatingclassifier\ C_3\right)$, corresponding to state 3.

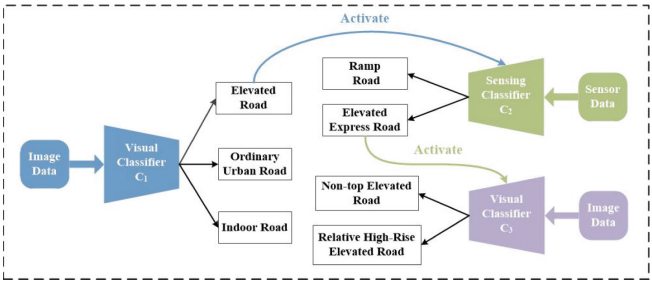


Fig. 5. The running process of three sub-classifiers.

Likewise, in the scene of the elevated expressway, if the current vehicle is about to exit the elevated road, i.e., the current section is a ramp section, corresponding to state 2, combined with the output of classifier C_1 and classifier C_2, it can be determined whether the vehicle is on a ramp section. After this, there are two possible scenarios. In scenario 1, the vehicle enters the elevated expressway located below (or above), corresponding to state 3, and the mechanism continues to run. In scenario 2, the vehicle enters an ordinary urban road, corresponding to state 1, and the process ends, with only classifier C_1 continuing to output. The specific schematic of this process is shown in Fig. 4 and Fig. 5.

Note that when the classifier is not activated, its output results are equiprobable. That is, when classifier C_2 is not activated, P^{2_1} and P^{2_2} are both equal to $\frac{1}{2}$; when classifier C_3 is not activated, P^{3_1} and P^{3_2} are both equal to $\frac{1}{2}$.

According to the above, using the visual images and sensor information collected synchronously during vehicle travel, it is possible to effectively distinguish between the four types of roads at different heights: underground roads, ground-level roads, ramp roads, and elevated expressways. In addition, visual image information can be used to distinguish between elevated expressways at different heights, further refining the granularity of the scene in elevated scene. At the same time, by combining with the dynamic driving process of the vehicle, it is possible to reduce the system error caused by the sub-classifier. That is, the

mutual switching of state 1 and state 3 must pass through state 2, which can be corrected when the classifier C_1 has an erroneous output.

5 Height Aware Map Matching Algorithm

The urban road driving scene classification model is applied to obtain a driving scene indicator matrix for each moment. A height aware map matching algorithm can be obtained by merging this matrix with the map matching algorithm. Currently, there are two types of map matching methods for GNSS trajectory data: global algorithms and local algorithms. However, regardless of the type of map matching algorithm, candidate paths are first searched in the map, and then the optimal matching path is selected through different methods. This paper takes this as a starting point. It uses the classic HMM algorithm (global algorithm) and Incremental algorithm (local algorithm) as examples to illustrate how to integrate the above scene classification model into the existing map matching algorithm, giving it the ability to perceive different heights of roads.

5.1 Height Aware Global Map Matching Algorithm

Currently, global algorithms are mostly derived from HMM map matching algorithm [9]. Based on HMM map matching, this is a decoding problem that seeks to find the hidden sequence that generates the observed sequence under certain observations. The hidden sequence represents the specific location of the vehicle equipped with a positioning device. In contrast, the observed sequence is a series of coordinate points generated by the GNSS trajectory produced by the positioning device.

This algorithm has multiple candidate paths for a GNSS point within a certain road network distance. Each candidate path is represented as a vertex in the Markov chain and is characterized by its observation probability. The candidate road segment set is defined as C. For the jth candidate road segment C_i^j in the candidate road segment set C_i of the ith sampled point P_i in the trajectory, its observation probability is calculated as follows:

$$P\left(C_i^j\right) = \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{(x_i^j)^2}{2\delta^2}} \cdot \boldsymbol{P}_{i,rtype}^j \tag{1}$$

where x_i^j is the vertical distance between the current sampling point t_i and the candidate road segment C_i^j, and δ is the GNSS positioning error.

The weights between adjacent GNSS candidate points are used as edges in a Markov chain and are represented by the transfer probabilities. The transfer probability V is calculated as:

$$V\left(C_{i-1}^l \to C_i^s\right) = \frac{d_{i-1\to i}}{w_{(i-1,l)\to(i,s)}} \tag{2}$$

where, the numerator denotes the spatial distance between two adjacent GNSS points; the denominator denotes the shortest road network distance between two adjacent candidate points.

Finally, the dynamic programming algorithm is used to find the optimal path in the road network that maximizes the product of observation probability and transfer probability as the final matching path. The specific solving process is shown in Algorithm 1.

GNSS points are more affected by noise in complex urban roads, and adding the driving scene indicator matrix makes the global map matching algorithm more robust in complex urban roads. At the same time, a more correct, comprehensive view of the trajectory can be obtained.

Algorithm 1: Dynamic Programming

Input: Trajectory $\boldsymbol{T} = \{\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n}\}$ and Urban Road Network $\boldsymbol{G}$.
Output: Matching Path $\boldsymbol{P'} = \{\boldsymbol{e'_1}, \boldsymbol{e'_2}, \ldots, \boldsymbol{e'_n}\}$.

Let $Func[\,]$ be the joint probability corresponding to the current state and $Prev[\,]$ be the previous state of the current state.
for *i=2 to n* **do**
  Get the sets of candidate paths C_{i-1} and C_i corresponding to sampling points t_{i-1} and t_i from G;
  for *each* C_i^s *in* C_i **do**
    `// Observation probability.`
    Calculate $P(C_i^s)$
    for *each* C_i^l *in* C_{i-1} **do**
      `// Initialization.`
      if *i=2* **then**
        Calculate $P(C_{i-1}^l)$
        $Func[C_{i-1}^l] = P(C_{i-1}^l)$
      end if
      `// Transfer probability.`
      Calculate $V_l^s = V(C_{i-1}^l \longrightarrow C_i^s)$
    end for
    $Func[C_i^s] = P(C_i^s) \cdot Max_{C_{i-1}^l \in C_{i-1}} \left\{Func[C_{i-1}^l] \cdot V_l^s\right\}$
    $r = Max_{C_{i-1}^l \in C_{i-1}} \left\{Func[C_{i-1}^l] \cdot V_l^s\right\}$
    $Prev[C_i^s] = r$
  end for
end for
`// Get the global optimal path endpoint.`
$End = Argmax_{C_n^k \in C_n} \left\{Func[C_n^k]\right\}$
$P'.append(End)$
for *i=2 to n* **do**
  `// Backtracking process.`
  $End = prev[End]$
  $P'.append(End)$
end for
$P'.reverse(\,)$
return $\boldsymbol{P'}$

5.2 Height Aware Local Map Matching Algorithm

The local algorithm must obtain a local optimal solution and gradually advance toward the global solution. The Incremental algorithm [7] uses a weighted calculation of the similarity of candidate paths based on the distance and angle difference. The distance measurement from the sampling point t_i to each candidate road segment C_i^j and the direction difference are defined respectively as:

$$S_d\left(t_i, C_i^j\right) = \mu_d - a \cdot d\left(t_i, C_i^j\right)^{n_d} \tag{3}$$

$$S_\alpha\left(t_i, C_i^j\right) = \mu_\alpha \cdot cos\left(\alpha_{i,j}\right)^{n_\alpha} \tag{4}$$

where μ and n are scale factors, a is the distance scale, and α is the angular difference.

Algorithm 2: Look-Ahead

Input: Trajectory $\boldsymbol{T} = \{\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n}\}$ and Urban Road Network $\boldsymbol{G}$.
Output: Matching Path $\boldsymbol{P'} = \left\{\boldsymbol{e'_1}, \boldsymbol{e'_2}, \ldots, \boldsymbol{e'_n}\right\}$.

for *i=1 to n-k* **do**
  Get the candidate sets $\{C_i, C_{i+1}, \ldots, C_{i+k}\}$ of sample point t_i and its next k sample points from G;
  for *each* C_i^j *in* C_i **do**
    `// Composite score.`
    Calculate S_i^j
    if $Type\left(Max\left(P_{i,rtype}^j\right)\right) == Type\left(C_i^j\right)$ **then**
      `// Conformance to the scene.`
      Calculate A_i^j
      `// Composite look-ahead score.`
    end if
  end for
  $e'_i = Max_{C_i^j \in C_i}\left\{A_i^j\right\}$
  $P'.append(e'_i)$
end for
`// Processing residual candidates.`
for *i=n-k to n* **do**
  for *each* C_i^j *in* C_i **do**
    Calculate S_i^j
  end for
  $e'_i = Max_{C_i^j \in C_i}\left\{S_i^j\right\}$
  $P'.append(e'_i)$
end for
return $\boldsymbol{P'}$

The urban driving scene indicators were integrated into the calculation of the composite results, defining the composite scores as follows:

$$S_{\Theta}\left(t_i, C_i^j\right) = (S_d + S_{\alpha}) \cdot \boldsymbol{P}_{i,rtype}^j \tag{5}$$

the higher the score of this metric, the better the match.

Finally, a look-ahead algorithm is used to calculate the total composite score after advancing k steps along all candidate road sections, from which the one with the highest similarity is selected as the matching result. The look-ahead score A is defined as:

$$A\left(t_i, C_i^j\right) = \sum_{k,l=0}^{depth} S_{\Theta}\left(t_{i+l}, C_{i,k}^j\right) \cdot \boldsymbol{P}_{i,rtype}^j \tag{6}$$

For the convenience, $S_{\Theta}\left(t_i, C_i^j\right)$ is shortened to S_i^j, $A\left(t_i, C_i^j\right)$ is shortened to A_i^j.

The specific solving process is shown in Algorithm 2.

With the addition of the driving scene indicator matrix, only candidate paths that match the driving scene are considered in the look-ahead process, which reduces unnecessary calculations and makes the local map matching algorithm converge faster.

5.3 Buffer Zones

All the fusion mechanisms mentioned above are based on absolute trust in the scene classification model. However, when vehicles switch between different road types while passing through the ramp road, there is an unavoidable delay and fluctuation in the classification results of the classifier. In order to ensure that the output results of driving scene indicator matrix do not mislead the process of map matching, therefore a particular buffer zone can be set before and after the road section passing through the ramp section, and the road section in this buffer zone will not be fused with the driving scene indicator matrix.

The situations where the road type is switched when passing through the ramp road are divided into two cases. Case 1: switching between ramp roads and ordinary city roads; Case 2: switching between ramp roads and elevated expressways. The experimental results of the above classifier show that the delay in case 1 is slightly smaller than in case 2. According to [13], the length of the ramp road is generally more than 150 m, with a speed limit of 60 km per hour. Therefore, the buffer zone for case 1 is set to 20 m, and the buffer zone for case 2 is set to 30 m.

6 Evaluation

6.1 Dataset

To evaluate the effectiveness of the mechanism proposed in this paper in real-world urban scenarios, OpenStreetMap data and three sets of driving trajectories

located in Shanghai [12] were selected for the experiments. Specific information can be found in Table 1 and Fig. 6. As OSM is a two-dimensional map and does not provide information on the number of layers of elevated roads, this attribute was manually added. Trajectories 1 and 2 were collected under normal conditions with good weather, suitable lighting, and clear images. Trajectory 3 was collected between 5–6 am, with soft lighting, poor weather conditions, and heavy rainfall throughout some sections, resulting in poor visibility. Trajectories 1–3 passed through complex urban road areas such as the Inner Ring Elevated Road, Middle Ring Elevated Road, Yixian Elevated Road, Dabaishu Interchange, Handan Road Tunnel, Outer Ring Interchange, Xinzhuang Interchange, and Gonghexin Road Interchange in Shanghai.

Table 1. Experimental Trajectory Specifics.

Parameter	T_1	T_2	T_3
Total Points	1441	1337	3695
Sampling Frequency $/s$	1	1	1
Track Length $/km$	24.89	23.73	73.91
Number of Road Sections	145	92	160
Light Conditions	Bright	Bright	Dim

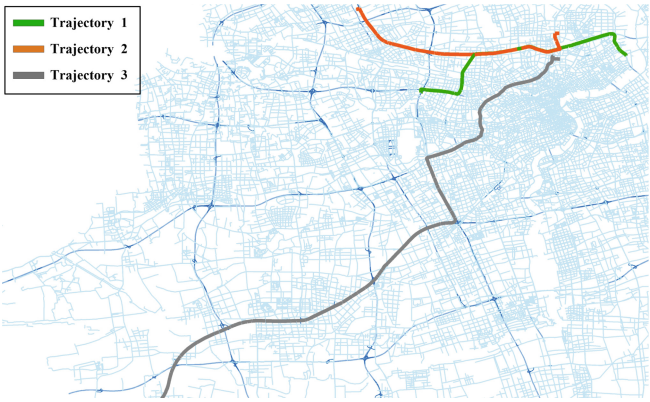


Fig. 6. Experimental Trajectory.

6.2 Map Matching Evaluation Methodology

In order to evaluate the improvement in performance of different types of map matching algorithms using this mechanism, the present paper selected the following three algorithms as baselines:

(1) The shortest distance map matching algorithm [14] (referred to as Point to Curve algorithm). This algorithm uses only the distance metric and chooses the path closest to the sampling point as the matching result.
(2) The local increment map matching algorithm [7] (referred to as Incremental algorithm). It uses a look-ahead algorithm to calculate the total similarity after moving k steps along all candidate paths, choosing the path with the highest similarity as the matching result.
(3) The Hidden Markov Model map matching algorithm [9] (referred to as HMM algorithm). The matching path with the highest global possibility is output by calculating the output probability and transition probability between adjacent states.

In this paper, match rate, precision, recall, and error rate are selected as the evaluation metrics for the algorithms.

Match rate is defined as the proportion of correctly matched points among all localized points in a trajectory [12]. For the HMM-based algorithms, since the algorithm output is a set of segments, the matching result for a single point is the nearest segment in the matching result.

$$Match\ Rate = \frac{N_{correct}}{N_{all}} \tag{7}$$

where $N_{correct}$ denotes the number of correctly matched positioning points with images and N_{all} denotes the total number of positioning points for the experimental trajectory.

Precision is defined as the percentage of the correct length of the matched path P' is calculated.

$$Precision = \frac{length\,(P \cap P')}{length\,(P')} \tag{8}$$

Recall is defined as the percentage of matching paths P' that cover the actual paths is calculated.

$$Recall = \frac{length\,(P \cap P')}{length\,(P)} \tag{9}$$

The above length (X) is the total length of the path X, not the number of edges. For example, length (P') is the length of the matching path P'.

The F_1 score is the summed mean of precision and recall, which combines precision and recall, and is defined as follows:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{10}$$

Using the F_1 score to define the error rate [6]:

$$Error\ Rate = 1 - F_1 \tag{11}$$

6.3 Driving Scene Classification Model

In order to give the model some degree of generalization, this paper collected driving recorder image data and sensor data from different vehicles under different weather conditions (sunny, light rain, moderate rain, heavy rain). Each image's urban driving scene category was manually labeled, and the sensor data under different elevated road scenes were extracted. The images of the elevated expressway were labeled again. 80% of the data was randomly selected as the training set, and the remaining 20% of the data and the images and sensor data of the three trajectories were set as the test set.

Since image data is relatively simple and easily distinguishable, and there is little correlation between sensor data, classifiers C_1 and C_3 used the ResNet-18 [15] for fine-tuning transfer learning. The input is image data, and specific data augmentation is performed, including random rotation, horizontal and vertical flipping, and Gaussian blur. Classifier C_2 used the SVM [16] for learning from scratch, with the input being vehicle speed, pitch angle, roll angle, and lateral angle information. Combining the dynamic running process of the vehicle and the test results of the three sub-classifiers, the calculated results obtained by the urban road driving scene classification model are shown in Table 2.

Table 2. Classification results of urban driving road Scenes.

Indicator	O	I	R	H	L
$\boldsymbol{P}$	98.8%	100%	96.7%	98.0%	98.3%
$\boldsymbol{R}$	90.7%	100%	94.1%	95.1%	95.4%
$\boldsymbol{F}_1$	94.5%	100%	95.4%	96.6%	96.9%

6.4 Performance on Complex Urban Roads

This subsection evaluates the performance of the map matching algorithm on complex urban road sections. To ensure that the candidate set contains all relevant roads, d in the Incremental and HMM algorithms is taken as $50m$, and other parameters are recommended in the original paper. The baseline method and the method incorporated with the mechanism M of this paper are applied to three trajectories and evaluated experimentally at different sampling intervals (1–10 s). The results are shown in Table 3 and Fig. 7.

According to Table 3, at sampling intervals of 1–10 s, the three baseline algorithms exhibited an average precision increase of 11.9%, 10.1%, and 10.9%, respectively, after incorporating the result information of the proposed urban driving scene model with spatio-temporal correlation in this paper. The average recall rates also increased by 7.5%, 7.8%, and 8.9%, respectively, while the average error rates decreased by 10.4%, 9.1%, and 10.1%. These indicate that in complex urban road scenarios, combining different map matching algorithms with the mechanism proposed in this paper can rectify mis-matchings in vertical road matching caused by differences in road height, thereby improving the matching performance of the algorithms.

Table 3. Average matching results of the 3 methods.

Method	Match Rate	Precision	Recall	Error Rate
Point to Curve	75.7%	62.4%	81.7%	29.5%
$M(P)$	**85.6%**	**74.4%**	**89.2%**	**18.9%**
Incremental	83.7%	77.9%	87.5%	17.7%
$M(I)$	**93.3%**	**88.1%**	**95.3%**	**8.5%**
HMM	83.9%	77.6%	87.5%	18.3%
$M(H)$	**93.7%**	**88.5%**	**95.5%**	**8.2%**

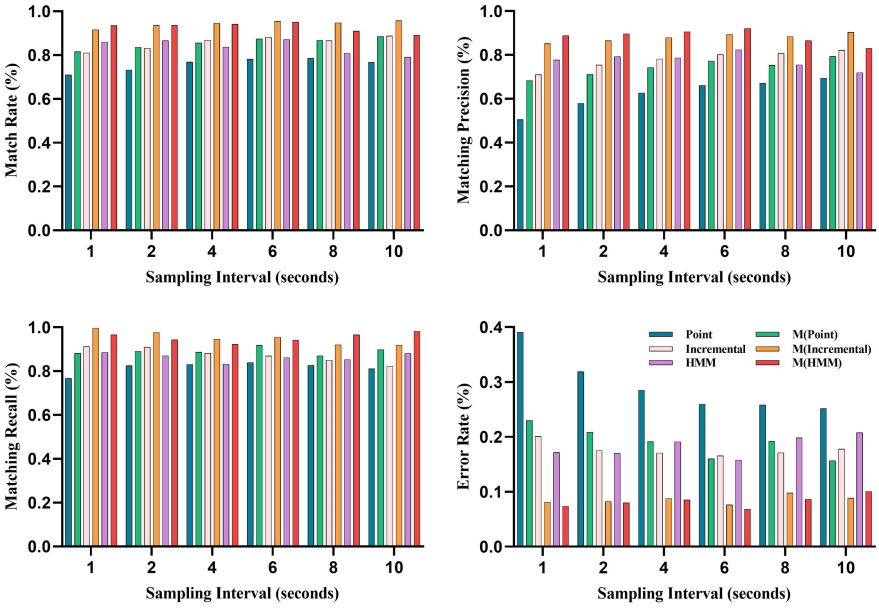
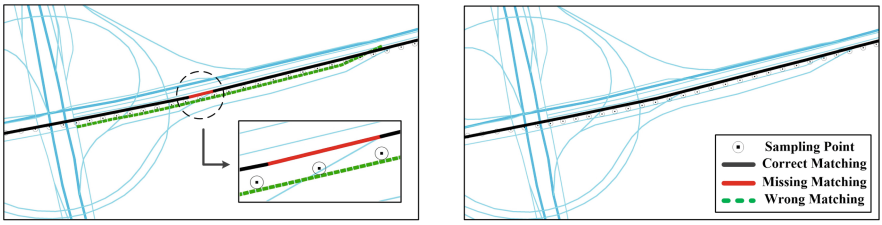


Fig. 7. Experimental results of the 3 methods at 1–10 s sampling interval.

Figure 8 compares the baseline algorithm and the boosted algorithm using the mechanism in the complex area. The fused mechanism proposed in this paper can better distinguish the elevated road from the parallel ground road, thus achieving better matching in the complex intersection and elevated road area. At the same time, the baseline algorithm has different degrees of wrong matching and missing matching.

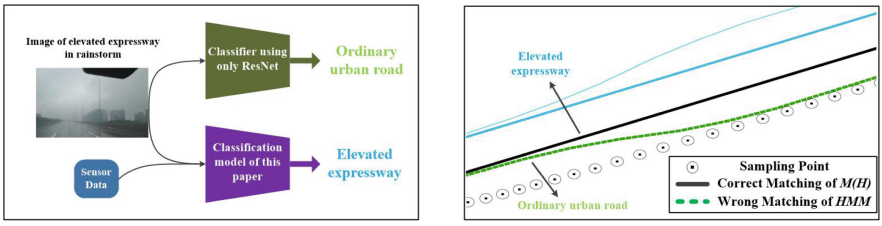
The mechanism proposed in this paper also provides the map matching algorithms with some resistance to weather effects. As shown in subplot (a) of Fig. 9, in the stormy road section of Trajectory 3, the vehicle images are blurred due to the severe weather conditions. At this time, it is very likely that the image of the elevated expressway will be classified as an ordinary urban road using only the image classifier of ResNet; while the classification model proposed in this

paper, which combines the different spatio-temporal states of the vehicle, can classify the image correctly. Since the classification model can transfer the previous correct classification results to the subsequent process, which reduces the impact of weather conditions on map matching. This situation is also reflected in the map matching algorithm. As shown in subplot (b) of Fig. 9, the map matching algorithm $M(H)$, which incorporates the mechanisms proposed in this paper, correctly matches the elevated expressway even in the stormy section of Trajectory 3; while the baseline algorithm HMM obtains an incorrect match.



(a) Incremental algorithm (b) Mechanism of this paper: $M(I)$

Fig. 8. The matching performance of the algorithm in complex urban roads with a sampling interval of 1s on Trajectory 2.



(a) Classification result (b) Matching result

Fig. 9. Demonstration of the results of different classifiers and different matching algorithms on a complex urban road with a sampling interval of 1s on Trajectory 3 under bad weather. Subplot (a) shows the recognition results of different classification models for elevated expressways under bad weather conditions. Subplot (b) shows the matching results between the map matching mechanism $M(H)$ in this paper and the baseline algorithm HMM.

6.5 Performance of Map Matching Algorithms

In terms of the performance of the map matching algorithm, the HMM algorithm, after incorporating the mechanism in this paper, maintains a better accuracy and lower error rate at different sampling rates, as shown in Fig. 7. Therefore, the mechanism of this paper makes the global algorithm more robust. The

Incremental algorithm, too, has improved the convergence speed by incorporating the mechanism of this paper, as shown in Table 4.

The performance of the Incremental algorithm is better at high sampling rates, as the local algorithm only obtains the matching section corresponding to the sampling point. As the distance between sampling points increases, there may be a situation where some intermediate sections are skipped, resulting in a decrease in performance as the sampling rate decreases. On the other hand, the HMM algorithm performs better at low sampling rates, assuming that each GNSS point in the global context is equally important. Due to noise in GNSS data, the error rate is higher when sampling frequently. As the distance between sampling points increases with increasing sampling intervals, a small number of GNSS points can provide a better overall direction of travel. Similarly, after incorporating information from the spatio-temporal correlated urban road scene classifier, the overall trend of map matching algorithms does not change at different sampling rates, as seen in Fig. 7. This suggests that the proposed mechanism relies on the baseline map matching algorithm, which improves the performance of different map matching algorithms, especially in complex urban sections such as elevated bridge sections, while still relying on the matching ability of the baseline algorithm itself in ordinary sections.

Table 4. Average single point matching time.

Sampling interval	1	2	4	6	8	10
Incremental ms	31.3	42.2	46.7	59.1	76.7	87.5
$\boldsymbol{M}$ (I) ms	**26.8**	**38.1**	**42.8**	**55.2**	**73.6**	**84.6**

7 Conclusion and Future Work

This paper proposes **HAUM^3**: A Height Aware Urban Map Matching Mechanism with Spatio-Temporal Correlation. This addresses the current lack of height information in map matching algorithms. On the one hand, the mechanism starts from the dynamic driving process of vehicles and designs a spatio-temporal correlation urban driving scene classification model based on the synchronized images and sensor data generated by vehicles, which can well distinguish five different heights of roads from a fine-grained level. On the other hand, according to the characteristics of different types of map matching algorithms, the classification model is integrated into traditional map matching algorithms, giving them height awareness. This improves the performance of the map matching algorithms and resists the influence of weather conditions to a certain extent. According to the evaluation results in the urban area of Shanghai, this mechanism significantly improves the accuracy of different map matching algorithms on complex urban roads, injecting new vitality into different traditional map matching algorithms.

There are still some ways to improve the mechanism in the future. First, this paper only distinguishes road scenes with different heights from the vertical direction, and the distinction of parallel roads is also an important issue. Second, future work could use image information as well as other on-board data to identify more specific scenes, such as side roads, sidewalks, etc. Thirdly, based on the road level matching achieved in this paper, lane level map matching methods for vehicles can also be further analyzed by using lane lines and other information in the images.

Acknowledgements. This work was supported by National Natural Science Foundation of China (62372188), GuangDong Natural Science Foundation (2021 A1515011755), as well as thanks to Yiren Technology Co. (Shanghai) for providing the experimental dataset.

References

1. Kubicka, M., Cela, A., Mounier, H., Niculescu, S.-I.: Comparative study and application-oriented classification of vehicular map-matching methods. IEEE Intell. Transp. Syst. Mag. **10**(2), 150–166 (2018)
2. GPS. Gps accuracy. https://www.gps.gov/systems/gps/performance/ (2020)
3. Ho, P.F., Hsu, C.C., Chen, J.C., Zhang, T.: Using barometer on smartphones to improve GPS navigation altitude accuracy. In: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, pp. 741–743 (2018)
4. Gong, Y., Zhu, Y., Yu, J.: Deel: detecting elevation of urban roads with smartphones on wheels. In: 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 363–371. IEEE (2015)
5. Zhang, X., Xu, H., Yang, J., Sun, J., Chen, F., Li, L.: Elevated road network: a metric learning method for recognizing whether a vehicle is on an elevated road. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2925–2932 (2020)
6. Li, H., Kulik, L., Ramamohanarao, K.: Robust inferences of travel paths from GPS trajectories. Int. J. Geogr. Inf. Sci. **29**(12), 2194–2222 (2015)
7. Brakatsoulas, S., Pfoser, D., Salas, R., Wenk, C.: On map-matching vehicle tracking data. In: Proceedings of the 31st International Conference on Very Large Data Bases, pp. 853–864 (2005)
8. Xiao, F., Zhang, J., Zhang, Y.: An online map matching algorithm based on second-order hidden markov model. J. Adv. Transp. **1–12**, 2021 (2021)
9. Newson, P., Krumm, J.: Hidden markov map matching through noise and sparseness. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 336–343 (2009)
10. Cui, G., Bian, W., Wang, X.: Hidden markov map matching based on trajectory segmentation with heading homogeneity. GeoInformatica **25**, 179–206 (2021)
11. Huang, Z., Qiao, S., Han, N., Yuan, C., Song, X., Xiao, Y.: Survey on vehicle map matching techniques. CAAI Trans. Intell. Technol. **6**(1), 55–71 (2021)
12. Fu, C., Huang, S., Tang, Y., Wu, H., Liu, C., Yao, L., Huang, W.: A real-time map matching method for road network using driving scenario classification. Acta Geodaetica et Cartographica Sinica **50**(11), 1617 (2021)

13. Baidu. Ramp (2023). http://baike.baidu.com/view/414889.htm?fr=aladdin
14. White, C.E., Bernstein, D., Kornhauser, A.L.: Some map matching algorithms for personal navigation assistants. Transp. Res. Part c: Emerg. Technol. **8**(1-6), 91–108 (2000)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
16. Chauhan, V.K., Dahiya, K., Sharma, A.: Problem formulations and solvers in linear SVM: a review. Artif. Intell. Rev. **52**(2), 803–855 (2018). https://doi.org/10.1007/s10462-018-9614-6