Optimization of Map Matching Algorithm in Various Road Conditions

Jingyang Liao*

School of Science, Southern University of Science and Technology, Shenzhen, 518055, China

*Corresponding Author Email: 11813113@mail.sustech.edu.cn

Abstract. Map matching refers to find the real driving trajectory of the vehicle by its GPS data. To solve this problem, most research focuses on improving the accuracy of GPS positioning. Even so, map matching still has a large deviation in complex road sections, such as interchanges, underground tunnels and other areas with high spatial complexity. At the same time, traditional map matching algorithms, such as hidden Markov model (HMM) and nearest neighbor algorithms, have the disadvantage of long running time. Different from the research on improving the accuracy of positioning, this research focuses on optimizing and creating a new algorithm for map matching. In order to solve the accuracy of map matching under complex road conditions, this paper adds a new attribute of altitude to GPS data for the first time, and greatly improves the accuracy of map matching in complex road conditions by constructing a three-dimensional hidden Markov model (3D HMM). In order to reduce the running time of the algorithm, this paper uses graph convolutional neural network as a bridge, integrates a variety of map matching algorithms and constructs a hybrid algorithm. Through experiments, it is found that the running time and accuracy of this hybrid algorithm are better than other algorithms.

Keywords: Map matching, HMM, 3D HMM, Hybrid algorithm.

1. Introduction

Map matching is to infer the vehicle's actual path by observed GPS data and known road networks. Nowadays, there are two key points in map matching problems, one is to improve the accuracy of relevant algorithms in complex terrain, and the other is to reduce the time complexity of relevant algorithms.

As everyone knows, the roads in many cities in the world are intricately distributed on a two-dimensional plane, such as New York, Tokyo and so on. Additionally, the roads in some cities like Chongqing are extremely complex in three-dimensional space. Google Maps, Apple Maps, and other mainstream maps often fail in these areas, and the actual location does not match the current location displayed on the map. At the beginning, pure geometric algorithms such as Nearest Neighbour algorithm were used to solve the map matching problem, but these pure geometric algorithms not only have a large amount of calculation and high time complexity, but also make mistakes in many cases [1]. With the proposal of hidden Markov model (HMM), Newson and Krumm used HMM to solve map matching for the first time in 2009. Different from the pure geometric algorithm, the HMM algorithm not only considers the current state of the system, but also considers the influence of the previous state on the current state. Based on this property of HMM algorithm, the accuracy of this method in complex road conditions has a qualitative leap compared with pure geometric algorithm [2, 3]. In 2013, based on HMM model, Ogami et al. applied Inverse Reinforcement Learning (IRL) to further optimize map matching algorithm on two-dimensional plane in complex situations [4]. In 2021, Fu et al. proposed an extended Viterbi algorithm and a self-adaptive sliding window mechanism, which is more accurate than the traditional HMM algorithm on complex road sections [5-7].

Most of the above studies are based on one algorithm, and these different algorithms are suitable for different road sections. They usually have good performance on a certain type of road section, but they are not so prominent on other road sections. For example, the Nearest Neighbour algorithm has good performance in areas with sparse roads, but it performs extremely poorly on complex road sections. On the contrary, HMM algorithm performs well on complex road sections, but its performance on sparse road sections is not as good as the Nearest Neighbour algorithm. This paper aims to study a hybrid algorithm, using different algorithms for map matching for different road sections. In addition, the above studies are all carried out on a two-dimensional plane. When vehicles enter underground tunnels or go up elevated roads, these traditional map matching algorithms will more or less fail. These studies did not consider the road matching problem in 3D space. Based on the traditional HMM algorithm, this paper proposes an optimized HMM algorithm suitable for 3D space.

First, this paper grided the map and divided it into $M \times N$ grids. Then, this paper used deep learning methods to identify the number of roads and intersections and whether there are elevated and underground tunnels in each grid [8]. This paper trained three different types of road segments using cross-validation. For each road segment, different algorithms are used for map matching to select the most suitable algorithm for each road segment. In addition, this paper conducted simulation experiments to select all the data in the grid where there are elevated or underground tunnels. Then this paper upscaled this part of the data, endow them with a new attribute of altitude, and randomly select values within an appropriate range. Finally, for this part of the data, this paper used the traditional HMM algorithm and the 3D HMM algorithm for map matching, and compare the advantages and disadvantages of the two.

2. Method

Through Fig. 1, the hybrid algorithm is divided into the following steps. First, the map of the area is gridded, divided into many small squares. The computer uses the graph convolutional neural network to identify the number of roads in each small square and whether there is an overpass, and then obtain a map information matrix of the area. At the same time, the GPS data obtained by the GPS system will be input into the computer for positioning to determine the position of the small grid where the vehicle is located. After obtaining the road information of the vehicle's location, the computer divides it into two categories by judging whether there is an overpass in the small square. If it exists, the system will further extract the altitude information of the vehicle to form three-dimensional data consisting of longitude, latitude, and altitude, and then bring it into the 3D HMM for map matching [9, 10]. If it does not exist, the computer divides it into two categories based on the number of roads. We trained the data set multiple times through the method of cross-validation, and finally set the division standard of the number of roads as 3. When the number of roads in the small grid is less than three, the computer will use a pure geometric algorithm for map matching. When the number of roads is greater than or equal to three, the computer will use traditional HMM for map matching.

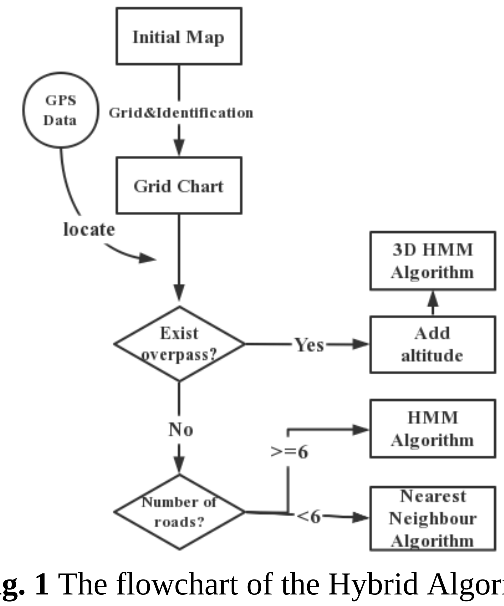

Fig. 1 The flowchart of the Hybrid Algorithm

2.1. Data Source

In order to study map matching under different road conditions, it is necessary to have a city with three road conditions: sparse roads, dense roads, and complex spatial structures. This article selects the Seattle city vehicle GPS dataset publicly released by Paul Newson and John Krumm in 2009 [2]. As the largest city in Washington State, Seattle has an extremely complex transportation network, as well as many complex spatial structures such as overpasses and underground tunnels. Therefore, the city of Seattle was an excellent choice for this research without doubt. The GPS dataset was sampled from Seattle, which contains a total of 7531 GPS data sampled at 1 Hz. The trip starts at (47.66748333, -122.1070833), which is shown in Fig. 2.

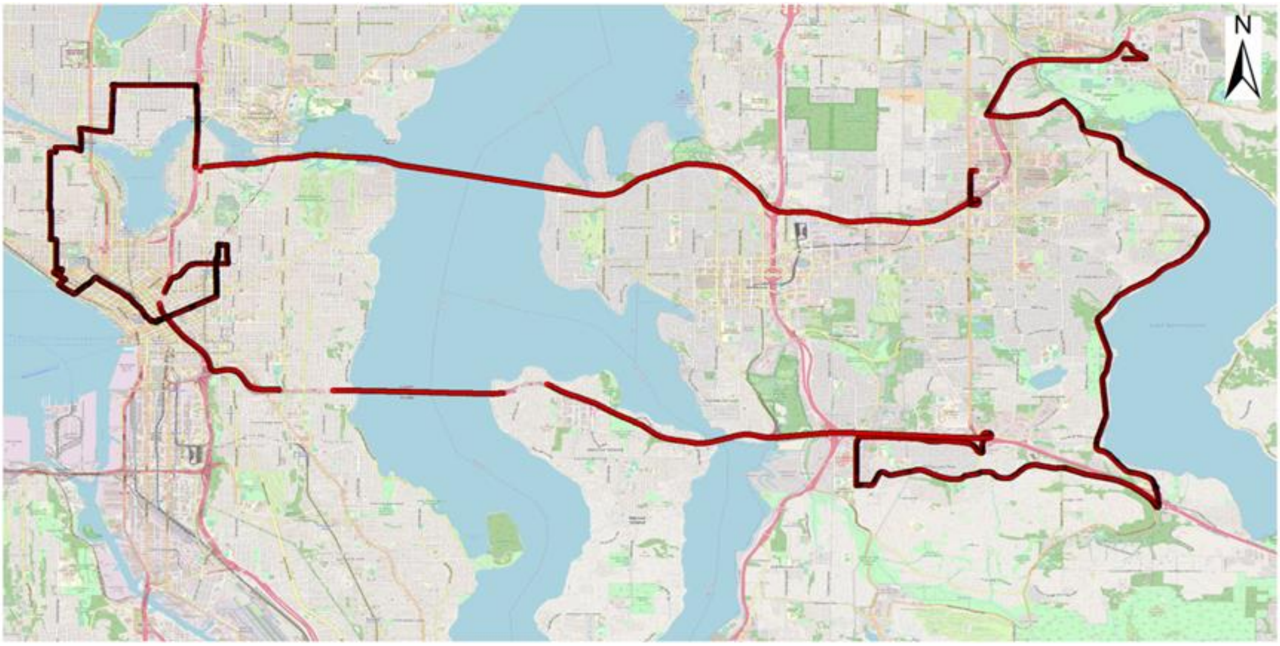

Fig. 2 Vehicle GPS data map

2.2. Data Preprocessing

The data set is actually a time series type data set, which contains many repetitions, which means that the vehicle stayed in place for this period of time, and this part of the data is not of great significance to this study, so this part of the repeated data was deleted. After this step of deletion, there are still 5991 data left.

In addition, this article marked a portion of the data as noise and removed them. When the distance between the current data and the previous data is greater than $2\sigma_Z$, this article think that the current data is caused by the movement of the vehicle, this data should be retained. The estimation of σ_Z will be presented in Section 2.4.3. Conversely, this paper believe that this difference is caused by noise rather than the movement of the vehicle, so we should delete this item of data. Through this step, we deleted 1,390 items of data, leaving 4,601 items of data in the end.

2.3. Problem Statement

As shown in Fig. 3, point A and point B are the positions of the GPS data collected by the computer, and Z_1 on the map is the real driving track of the vehicle. Map matching is to find the most likely vehicle trajectory based on the received GPS data. For point A, at this moment, the real position of the vehicle may be in Z_1, Z_2 and Z_3. Based on Nearest Neighbour algorithm, we think that point A is most likely to be on Z_1 because it is the closest to Z_1. However, when we use the same method to locate point B at the next moment, we will find that point B is most likely to be on Z_3, which is obviously impossible [11]. Therefore, the map matching problem can be equivalent to finding an optimal path in the case of known observation points (GPS data), so that the probability of observing this sequence of GPS data under this path is the largest. We use $\mathbf{X}_{1:T}$ to represent all the GPS data observed from 1 to T, and $\mathbf{Y}_{1:T}$ to represent the real position of the vehicle from 1 to T. The problem can be expressed as

$$arg\ max_{\boldsymbol{Y}_{1:T}} p(\mathbf{Y}_{1:T} \mid \boldsymbol{X}_{1:T}) \tag{1}$$

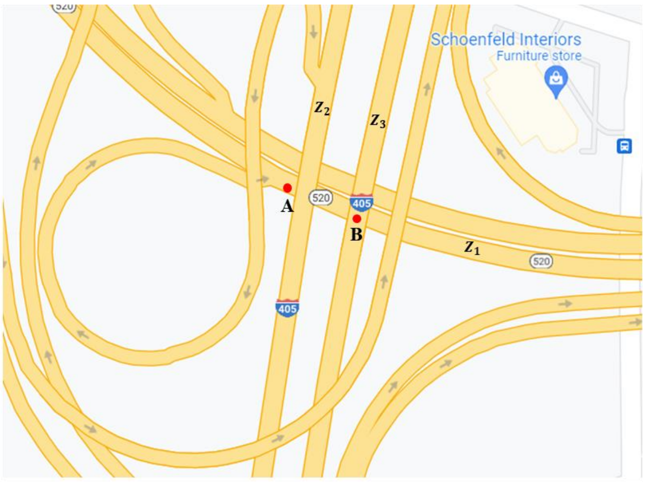

Fig. 3 Some GPS points on complex roads

2.4. Hidden Markov Model

As shown in Figure 4, HMM is actually a hidden Markov process, which is often used to solve speech recognition and map matching. When we use HMM to solve map matching, the real position of the vehicle at a certain moment is the latent variable Y, and the value space of Y is $\{r_1, r_2, \ldots, r_N\}$. The GPS data is the observed value $\mathbf{X} = (x_1, x_2)$, x_1 and x_2 are the two attributes of longitude and latitude in the GPS data. The value space of $\mathbf{X}$ is $\boldsymbol{\Theta} = (\Theta_1, \Theta_2, \ldots, \Theta_S)$, which is the Seattle Area. In map matching, we need to derive Y from $\mathbf{X}$, which is a decoding problem in HMM [12]. The key to solve this decoding problem is to find the parameter $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ which is shown in Equation (2).

$$\begin{cases} p(Y_{t+1} = r_j | Y_t = r_i) = a_{ij}, 1 \le i, j \le N, \mathbf{A} = [a_{ij}]_{N\times N} \\ p(\mathbf{X_t} = \mathbf{x_t} | Y_t = r_i) = b_{ij},\ 1 \le i \le N, 1 \le j \le S, \mathbf{B} = [b_{ij}]_{N\times S} \\ P(Y_1 = r_i) = \pi_i, 1 \le i \le N, \boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N) \end{cases} \tag{2}$$

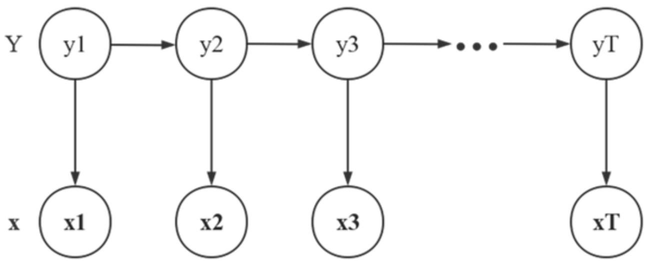

Fig. 4 The flowchart of HMM

2.4.1 Emission probability

In map matching, the emission probability is used to measure the probability of observing $\mathbf{x_t}$ when the vehicle is on road segment r_i, which can be denoted as $p(\mathbf{X_t} = \mathbf{x_t} | Y_t = r_i)$ in Equation (2).

Based on research by VanDiggelen et al., This paper considers that the GPS noise, that is, $\|\mathbf{x_t} - \mathbf{r_{t,i}}\|$ shown in Fig. 5, approximately obeys a normal distribution with a mean value of zero [2]. We have

$$p(\mathbf{X_t} = \mathbf{x_t} | Y_t = r_i) = \frac{1}{\sqrt{2\pi}\sigma_Z} e^{-\frac{\|\mathbf{x_t} - \mathbf{r_{t,i}}\|^2}{2\sigma_Z{}^2}} \tag{3}$$

2.4.2 Transition probability

As shown in Fig. 5, the transition probability $\mathrm{p}(\mathrm{Y}_{t+1} = \mathrm{r}_j|\mathrm{Y}_t = \mathrm{r}_i)$ is the probability that the vehicle moves from $\mathbf{r}_{t,i}$ to $\mathbf{r}_{t+1,j}$. Based on research by Newson and Krumm, $\left|\|\mathbf{x}_{t+1} - \mathbf{x}_t\| - \|\mathbf{r}_{t+1,j^*} - \mathbf{r}_{t,i^*}\|\right|$ approximately follows an exponential distribution with a mean value of μ [2]. The estimation of μ will be presented in Section 2.4.3. We have

$$\begin{cases} \mathrm{p}(\mathrm{Y}_{t+1} = \mathrm{r}_j|\mathrm{Y}_t = \mathrm{r}_i) = \frac{1}{\mu} e^{-\frac{\left|\|\mathbf{x}_{t+1}-\mathbf{x}_t\|-\|\mathbf{r}_{t+1,j^*}-\mathbf{r}_{t,i^*}\|\right|}{\mu}} \\ \mathrm{E}(\mathrm{Y}_{t+1} = \mathrm{r}_j|\mathrm{Y}_t = \mathrm{r}_i) = \mu \end{cases} \tag{4}$$

Where i^* and j^* respectively reflect the nearest road segments at time t and t+1.

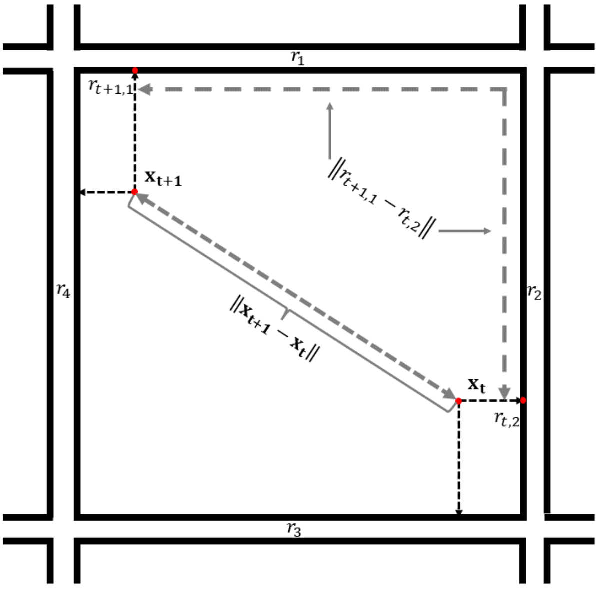

Fig. 5 An explanation for some distance

2.4.3 Parameter estimation

There are two parameters in HMM to be estimated here. One is σ_Z, which is the standard deviation of GPS noise. Based on the definition of standard deviation and Equation (3), we use Equation (5) to estimate σ_Z.

$$\widehat{\sigma_Z} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{r}_{t,i^*}\|^2} \tag{5}$$

The other parameter is μ. Based on the properties of the exponential distribution and Equation (4), we use Equation (6) to estimate μ.

$$\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T}\left|\|\mathbf{x}_{t+1} - \mathbf{x}_t\| - \|\mathbf{r}_{t+1,j^*} - \mathbf{r}_{t,i^*}\|\right| \tag{6}$$

2.5. The 3D Hidden Markov Model

Fig. 6 intercepts part of the dataset used in this article, and this part of the data is in the overpass section. In this study, inaccurate positioning often occurs when using traditional HMM for map matching, and this deviation often appears in the vertical plane instead of the horizontal plane. Based on the above points, this paper considers that it is caused by the factor of altitude. Since the dataset does not contain the attribute of altitude, and it is difficult to find a suitable vehicle GPS dataset with the attribute of altitude. This paper adopts the method of simulation experiment and gives a new attribute h to the $\mathbf{x} = (x_1, x_2)$ mentioned above to make it $\mathbf{x} = (x_1, x_2, h)$. The meaning of h is the altitude of the vehicle at time t. Then this paper brings the updated x into the HMM in Section 2.4 for map matching in three-dimensional space.

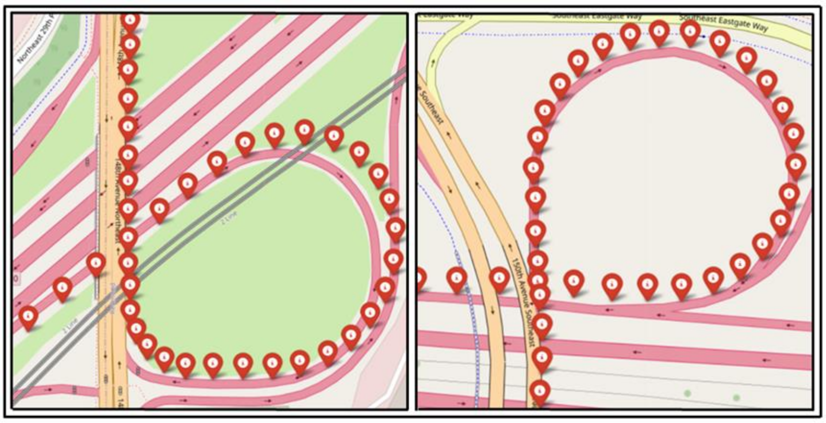

Fig. 6 Some examples of GPS points on overpasses

3. Results and Discussion

3.1. Evaluation Standard

This paper use recall to measure the ability of the model to correctly predict samples, which is equal to the ratio of the length of the correctly matched road segments to the total length of the real trajectory. We have

$$\text{Recall} = \frac{\text{length of the correctly matched road segments}}{\text{length of the real trajectory}} \tag{7}$$

This paper use precision to measure the accuracy of the model prediction, which is equal to the ratio of the length of the correctly matched road segments to the total length of the matched trajectory. We have

$$\text{Precision} = \frac{\text{length of the correctly matched road segments}}{\text{length of the matched trajectory}} \tag{8}$$

In order to reduce the loss of information, this paper adopts $\mathrm{F_1 - score}$ to evaluate the pros and cons of the model. $\mathrm{F_1 - score}$ is between 0 and 1. The model will perform better with larger $\mathrm{F_1 - score}$. $\mathrm{F_1 - score}$ can be expressed as:

$$\mathrm{F_1 - score} = \frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}} \tag{9}$$

3.2. Road Density Map

In order to find the most suitable classification criteria for regional categories, this article uses a hybrid algorithm for map matching on the data set under different k values. Among them, k represents the division standard between the nearest neighbor algorithm and the HMM model. When the number of roads in the area is less than k, the nearest neighbor algorithm is used, otherwise, the HMM is adopted. The paper compares the three evaluation standards mentioned in section 3.1 in various scenarios on this dataset. Based on Fig. 7, this research finds that when k is less than or equal to 6, the F-score of the hybrid algorithm tends to be stable. This paper believes that this is because the area with the number of roads less than 6 is very rare, so the hybrid algorithm is basically equivalent to HMM or 3D HMM at this time. When k is greater than or equal to 6, the area that meets this condition increases rapidly. In these areas, using nearest neighbor algorithm will greatly reduce running time, although the accuracy of map matching will decrease to some extent, nearest neighbor algorithm is still a good choice. When the number of k is greater than 9, there is almost no region where the number of roads is greater than 9, and the hybrid algorithm is equivalent to the nearest neighbor algorithm. In summary, this paper believes that the mixed model performs best when k is equal to 6. At this time, the hybrid model can meet the requirements of map matching with a short running time while ensuring high accuracy.

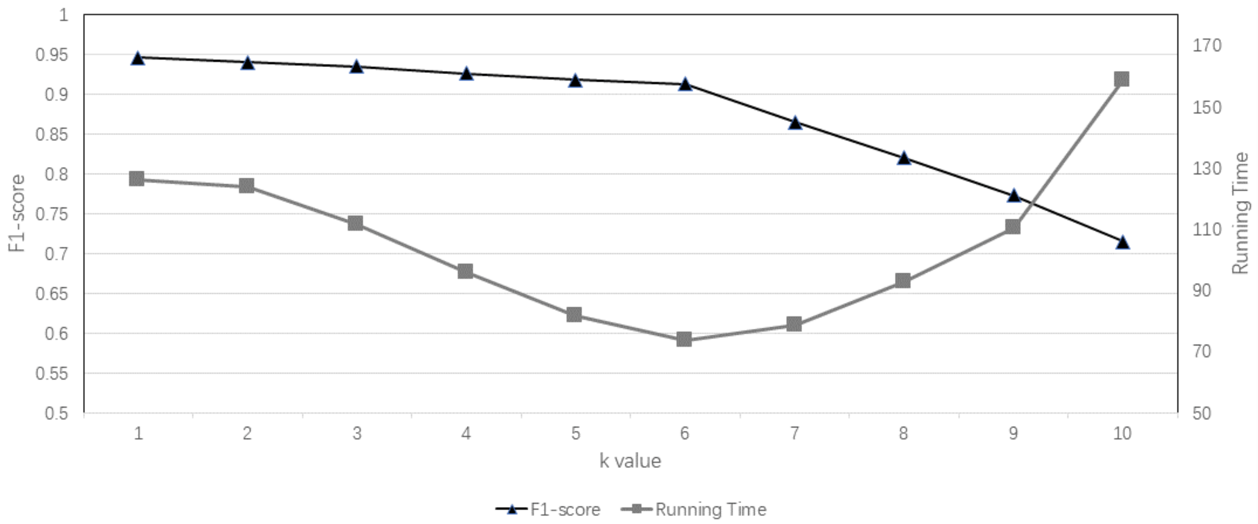

Fig. 7 Determination of k value

After obtaining the map of Seattle, this article divides the city of Seattle into 39×21 square areas. As shown in Fig. 8, this study uses a convolutional neural network algorithm for image recognition in each square area to obtain the number of roads and whether there is an overpass.

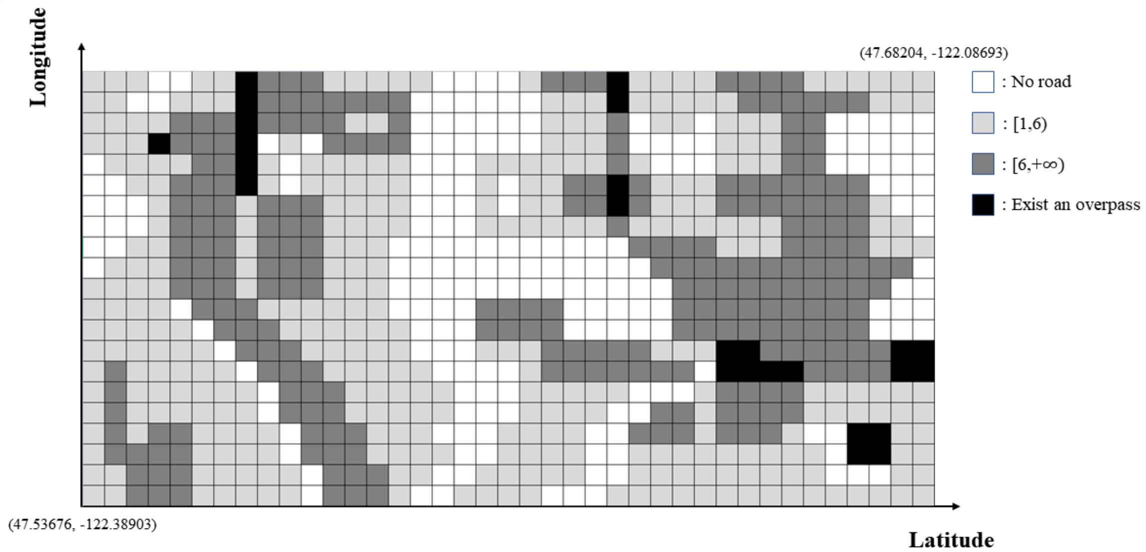

Fig. 8 Road Density Map of Seattle

3.3. Model Evaluation

Based on Equation (7), (8) and (9), this article uses the dataset to evaluate the four models mentioned above. We can find from Fig. 9 and Table 1 that for the prediction of the entire dataset, precision, recall and $F_1 - \text{score}$ of 3D HMM are the highest, and its prediction effect is the best. In addition, although the indicators of the HMM and the hybrid model are not as good as 3D HMM, the difference among them is very small. In terms of calculation speed, Nearest Neighbor is the fastest, followed by hybrid model, and HMM and 3DHMM are slower than the above two models.

Table 1. Comparison of the four models

Model	Recall	Precision	$F_1 - \text{score}$	Running Time (s)
Nearest Neighbour	0.715	0.692	0.703	192.7
HMM	0.917	0.895	0.906	106.9
3D HMM	0.955	0.946	0.950	169.9
Hybrid Model	0.925	0.902	0.913	73.5

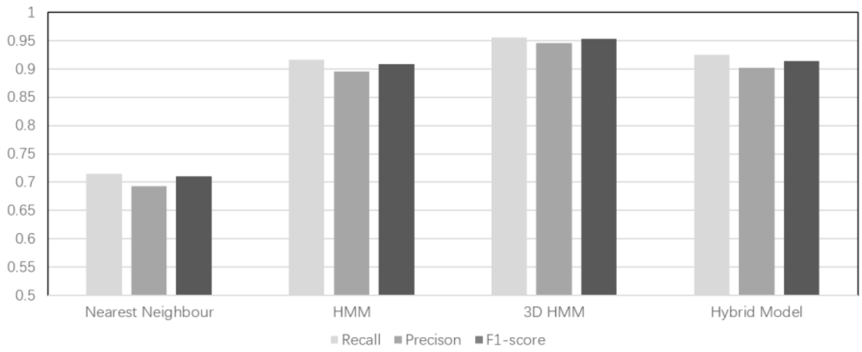

Fig. 9 Bar chart of the four models

4. Conclusion

This research aims to optimize the existing classic map matching algorithm, and greatly improve the running speed while ensuring the accuracy. To improve the accuracy of existing map matching algorithms on spatially complex road sections, this study uses the altitude data of vehicles for auxiliary positioning for the first time, which greatly increases the accuracy of map matching on this road section. In addition, the hybrid algorithm proposed for the first time in this study not only greatly improves the accuracy of map matching, but also greatly reduces its running time compared to HMM. Although the running time of the hybrid algorithm is greatly reduced compared to other algorithms, it still has a lot of room for improvement. This study spends a lot of time using convolutional neural networks for map gridding and recognition. With the development of deep learning, the algorithm for image recognition will be more efficient, and the running time will be further shortened.

References

[1] Quddus M. A, Ochieng W. Y, Noland R. B. Current map-matching algorithms for transport applications: state-of-the art and future research directions. Transportation Research Part C Emerging Technologies, 2007, 15(5), 312-328.

[2] Newson P, Krumm J. Hidden Markov map matching through noise and sparseness. 17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2009.

[3] Yuan J, Zheng Y, Zhang C, Xie X, Sun G. Z. An Interactive Voting-based Map Matching Algorithm. Journal of Signal Processing Systems, 2010.

[4] Osogami T, Raymond R. Map matching with inverse reinforcement learning. International Joint Conference on Artificial Intelligence, 2013.

[5] Chu H. J, Tsai G. J, Chiang K. W, Duong, T. T. Gps/mems ins data fusion and map matching in urban areas. Sensors, 2013.

[6] Bojja J, Kirkko-Jaakkola M, Collin J, Takala J. Indoor localization methods using dead reckoning and 3D map matching. Journal of Signal Processing Systems, 2014, 76(3), 301-312.

[7] Kalchbrenner N, Grefenstette E, Blunsom, P. A convolutional neural network for modelling sentences. Eprint Arxiv, 2014, 1.

[8] Quddus M, Washington S. Shortest path and vehicle trajectory aided map-matching for low frequency gps data. Transportation Research, 2015, 55, 328-339.

[9] Aly H, Youssef M. Semmatch: road semantics-based accurate map matching for challenging positioning data. ACM, 2015.

[10] Hsueh Y L, Chen H C. Map matching for low-sampling-rate GPS trajectories by exploring real-time moving directions. Information Systems, 2016, 55-69.

[11] Mohamed R, Aly H, Youssef M. Accurate real-time map matching for challenging environments. IEEE Transactions on Intelligent Transportation Systems, 2016, 18(4): 847-857.

[12] Fu X, Zhang J, Zhang Y. An online map matching algorithm based on second-order hidden Markov model. Hindawi Limited, 2021.