

Research Article

Lane-Level Road Map Construction considering Vehicle Lane-Changing Behavior

Liang Fan ^{1,2}, Jinfen Zhang,^{1,2} Chengpeng Wan,^{1,2} Zhongliang Fu,³ and Shiwei Shao ⁴

¹Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430070, China

²Inland Port and Shipping Industry Research Co., Ltd. of Guangdong Province, Shaoguan 512000, China

³School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China

⁴National-Local Joint Engineering Laboratory of Geo-Spatial Information Technology, Hunan University of Science and Technology, Xiangtan 411201, China

Correspondence should be addressed to Shiwei Shao; shaoshiwei@163.com

Received 26 April 2022; Revised 20 November 2022; Accepted 1 December 2022; Published 21 December 2022

Academic Editor: Jaeyoung Lee

Copyright © 2022 Liang Fan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the construction of lane-level road maps has received extensive attention from industry and academia. It has been widely studied because this kind of map provides the foundation for much research, such as high-precision navigation, driving behavior analysis, and traffic analysis. Trajectory-based crowd-mapping is an emerging approach to lane-level map construction. However, the major problem is that existing methods neglect modeling the trajectory distribution in the longitudinal direction of the road, which significantly impacts precision. Thus, this article proposes a two-stage method based on vehicle lane-changing behavior to model the road's lateral and longitudinal trajectory distributions simultaneously. In the first stage, lane-changing behaviors are extracted from vehicle trajectories. In the second stage, the lane extraction model is established using the weighted constrained Gaussian mixture model and hidden Markov model to estimate lane parameters (e.g., lane counts and lane centerline) on each road cross section. Then accurate and continuous lane centerlines can be constructed accordingly. The proposed method is verified using vehicle trajectory data collected from the crowdsourced platform named Mapillary. The results show that the proposed method can construct lane-level road information satisfactorily.

1. Introduction

Lane-level road maps are the core of automatic driverless systems and intelligent assisted driving systems. It can be used for autonomous vehicle navigation and online driver guidance. Besides, such a map can also be important for lane-based traffic analysis. The analysis of the traffic flow will be done more accurately if it is based on a lane-level map that is fully compatible with the input trajectory data. Currently, lane-level road maps are usually acquired from high-resolution remote sensing images, vehicle-mounted laser point clouds, or differential GPS trajectories with an accuracy of about 0.5–4 m [1]. These manual and semimanual approaches are time-consuming and labor-intensive.

Crowdsourcing is a low-cost and efficient way to extract useful road information from data acquired by crowd

participants or volunteers. The crowdsourced method has been successfully applied in road map construction, which collects road data by crowding vehicles [2–4]. The advantage of crowdsourced technology has led to a relatively limited number of scholarly papers on lane-level map construction based on crowdsourced trajectories. However, these efforts mainly model the trajectory distribution on road cross sections and do not model the trajectory distribution in the longitudinal direction of the road. It can cause the lane counts on a road to not be constant throughout a segment of the road, while road designers usually maintain the road width constant throughout the segment.

This paper proposes a strategy to model the road longitudinal trajectory distribution by considering vehicle lane-changing behavior. The method is based on the observation that the probability of a vehicle changing lanes increases on

road sections where the number of lanes changes. Thus, the lane-changing behaviors of vehicles are first identified from their trajectories. Then, we use the weighted constrained Gaussian mixture model (WCGMM) and the hidden Markov model (HMM) to establish the relationship between the vehicle lane-changing behavior and the number of lanes on the adjacent road cross sections. At last, each road segment's lane counts and lane centerlines are extracted. In summary, the contributions of this paper are the following:

- (1) We propose a vehicle lane-changing behavior extraction method based on the clustering of trajectory points.
- (2) We propose a lane-level road map construction method based on the simultaneous modeling of the trajectory distribution in a road's lateral and longitudinal directions.

The rest of this article is structured as follows: Section 2 gives an overview of works related to this article. Section 3 describes the lane-changing behavior recognition method and the lane centerline extraction approach based on lane-changing behavior. Experiments and results are discussed in Section 4, followed by conclusions and future work in Section 5.

2. Related Work

Crowdsourced vehicle trajectory data is a low-cost, real-time data source that potentially contains rich road information. Thus, many scholars have studied building a navigation map using crowdsourced vehicle trajectories [5–8]. With the rise of autonomous driving technology, high-definition navigation maps have become one of the critical capabilities for autonomous driving. It can provide vehicles with more reliable environmental perception capabilities. Compared with traditional navigation maps, high-definition maps for autonomous driving require more real-time and accurate lane information. Constructing lane-level road maps using crowdsourced vehicle trajectories has also become a new research hotspot.

Generally, the lane-level road map construction method based on crowdsourced vehicle trajectories usually includes three steps. First, the noise of raw trajectories is filtered. For example, previous studies have used a Kalman filter and a particle filter algorithm [9] or kernel density methods [10] for trajectory data preprocessing. Second, the lane counts of the segment are estimated. In previous studies, Edelkamp and Schrödl [11] proposed a K-means clustering method to extract lane counts from massive differential GNSS trajectories to construct and update urban road digital maps. Uduwaragoda et al. [12] used nonparametric Kernel Density Estimation (KDE) to estimate the number and location of the lane centerlines. Chen and Krumm used vehicle trajectories recorded from standard GPS devices to construct lane-based, routable digital maps [13]. They used perpendicular lines at certain distances relative to the road's centerline for one-dimensional classification of the trajectories. Then, prior knowledge and the constrained Gaussian mixture model (CGMM) are used to get better classification results. Tang

et al. [14] proposed a naive Bayesian classifier to extract the number and the rules of traffic lanes. In addition, an optimized CGMM was proposed to mine the traffic lanes' numbers and locations [15]. Furthermore, a fuzzy-set-based algorithm is proposed to construct the lane geometry near road intersections using traffic rules [16]. Roeth et al. [17] proposed a method based on elementary building blocks that guarantees applicable lane models and used a reversible jump Markov chain Monte Carlo method to explore the model parameters. Third, these lane models of road segments are connected into lanes. Gong et al. proposed a region-growing cluster algorithm with distance and orientation constraints to construct lane-level roads [18]. Arman and Tampère [19, 20] first identify road nodes and divide the network into segments. Then they construct lanes for each segment and connect these lanes. Zheng et al. [21] summarize the lane-level road geometry extraction methods and mathematical modeling of a lane-level road network. They analyzed these two parts' methodologies, advantages, and limitations and discussed the classic logic formats of a lane-level road network.

It can be found that lane count extraction is a critical step in lane construction, according to the above research methods. GMM uses prior knowledge of lanes. Thus, the methods based on GMM have higher robustness for low-precision trajectory data. However, the influence of vehicle lane-changing behavior is not considered in the GMM method, which leads to the differences between the actual trajectory distribution and the Gaussian mixture distribution. In addition, researchers usually use a one-dimensional classification method to extract the number of lanes. It can lead to variable lane width over the whole road segment. Road designers typically keep the number and width of lanes constant over a section of road. The results revealed that there was still room to improve the accuracy and precision of the lane geometry [22], although the crowdsourced trajectory method was economical.

3. Methodology

We propose a two-step algorithm that converts a set of trajectory data into a lane-level road map. Figure 1 gives an overview of the method, which is detailed hereafter. The first step of the algorithm is vehicle lane-changing behavior recognition. At the end of this step, the lane-changing behaviors will be identified from vehicle trajectories. We select the trajectory segments with lane-changing behaviors to model the change of lane counts between adjacent road cross sections. In the second step, the lane counts and lane centerline nodes are extracted, and we connect these nodes to construct lane centerlines for each unidirectional road segment.

3.1. Recognition of Lane-Changing Behavior. The recognition of lane-changing behavior is explained in the following sections.

3.1.1. Lane-Changing Feature Extraction. Driving behavior can be divided into car-following and lane-changing behavior [23]. The car-following behavior refers to following

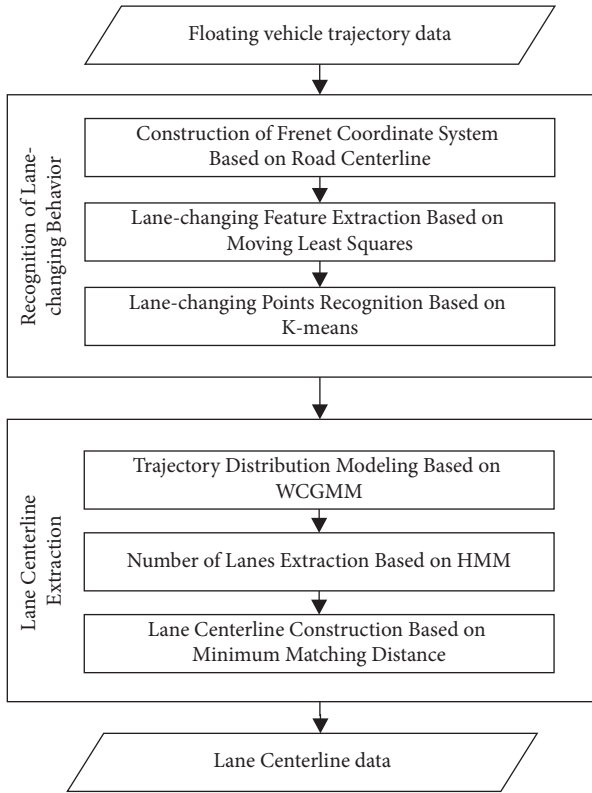


FIGURE 1: Flowchart of lane information extraction.

the preceding vehicle in the same lane. The lane-changing behavior refers to the vehicle entering an adjacent lane to satisfy its driving purpose. In this paper, the trajectory with the former behavior is called the car-following trajectory segment, and the latter is called the lane-changing trajectory segment. It is clear that the points on a car-following trajectory segment are located in the same lane, and the points on a lane-changing trajectory segment will cross the adjacent lanes.

We divide the vehicle trajectories into car-following and lane-changing trajectory segments in the Frenet coordinate system. Compared with other coordinate systems, the Frenet coordinate system can distinguish the vehicle's motion state on the road from the geometric shape of the road itself. The Frenet coordinate system converts the two-dimensional motion of a vehicle in plane space into two one-dimensional motions in reference curve space. One-dimensional motion problems are more accessible to model and resolve than two-dimensional ones.

Unlike the conventional Cartesian coordinate system, the Frenet coordinate system needs to be based on a given reference line to convert the absolute position of an object to the relative position of a reference line. The road centerline is selected as the reference line in a Frenet coordinate system, and the vehicle trajectories in the coordinate system are converted to the relative positions of the road centerline. The new trajectory points can directly reflect the state of vehicle motion on the road without being affected by road geometry.

In the Frenet coordinate system, lateral and longitudinal displacements are used to represent the position of a vehicle.

Assuming that the coordinates of a vehicle in a global coordinate system (Cartesian coordinate system or Geographic coordinate system) are $P_t(x, y)$. The projection is made from point P_t to road centerline T_{ref} . P_s is the projection point. The distance between P_s and P_t is d . The curve distance from the starting point of T_{ref} to P_s is s . d is the lateral displacement between the vehicle and the road centerline. s is the vehicle's longitudinal displacement along the road centerline's extension direction, as shown in Figure 2. Therefore, the vehicle's coordinates in the Frenet coordinate system are $F_r(s, d)$. Figure 3(a) shows the vehicle's trajectory in Figure 2 using the Frenet coordinate system. Compared with the trajectory in the global coordinate system (Figure 3(b)), the shape of the vehicle trajectory curve in the Frenet coordinate system is more straightforward and intuitive. The ordinate of the curve can directly reflect the position information of the lane where the vehicle is located. Thus, it is easy to detect the lane-changing behavior of a vehicle in the Frenet coordinate system.

In the Frenet coordinate system, the car-following trajectory segment is a horizontal straight-line segment. The lane-changing trajectory segment is an "S"-shaped curve that crosses the adjacent lanes. Compared with the lane-changing trajectory segment, the lateral displacement of points on the car-following trajectory segment is similar, and the slope of each point is around zero. Therefore, they can be distinguished according to the lateral displacement and slope of the trajectory points. We select two indices to describe lane-changing features of trajectory points. One is the difference between the lateral displacement of neighborhood points. The other is the slope at the target trajectory point.

Because a consumer-grade GPS equipped on a floating car usually has a significant positioning error, lane-changing features extracted by vehicle trajectory points are usually affected by the positioning error. Therefore, we use the moving least squares method [24] to fit the actual trajectory of a vehicle before extracting the lane-changing features. Compared with the original trajectory, the fitted trajectory is smoother and less affected by positioning errors. Lane-changing features extracted based on the fitted trajectory are more stable.

3.1.2. Recognition of Lane-Changing Trajectory Points.

After extracting the lane-changing features of each trajectory point, lane-changing trajectory points can be classified based on the features. We use the K -means algorithm to classify points and recognize lane-changing behavior for each trajectory. K -means is a common clustering algorithm widely used because of its simplicity and high efficiency. We chose this algorithm because the number of points on each trajectory is not very large and the number of clusters is determined. It should be noted that this paper mainly focuses on the accuracy of lane information extraction and does not consider the speed of algorithm execution. The distributed clustering algorithm can also be used to replace the traditional K -means algorithm, considering the algorithm's efficiency. The lane-changing point recognition method based on K -means includes the following three steps:

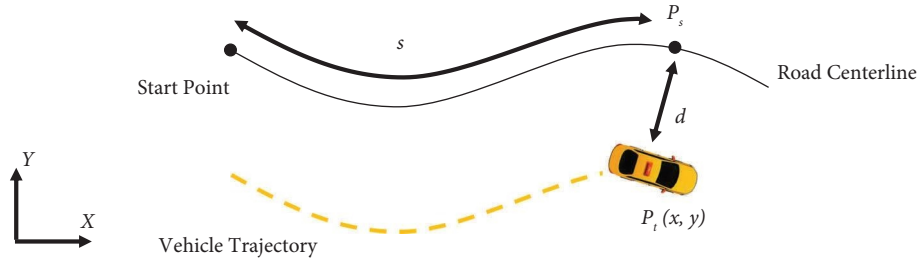


FIGURE 2: Conversion from global coordinate system to Frenet coordinate system.

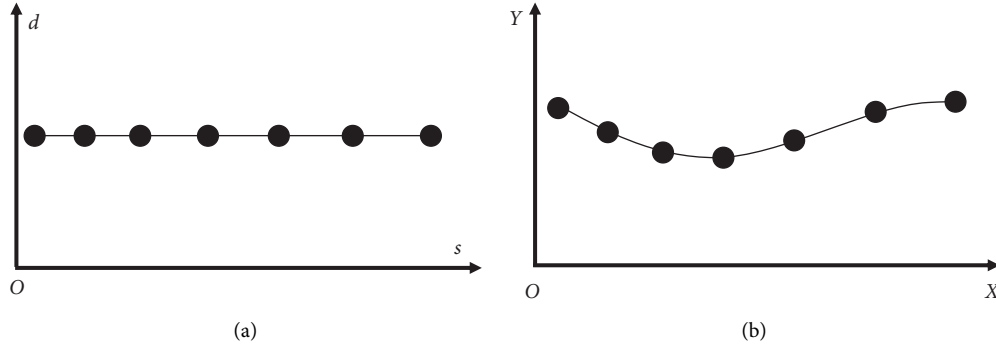


FIGURE 3: Representation of vehicle trajectory curve in different coordinate systems: (a) vehicle trajectory in the Frenet coordinate system; (b) vehicle trajectory in the global coordinate system.

Step 1: Normalization of lane-changing eigenvalues. Since the two extracted lane-changing features are not uniform in dimension and the numerical ranges of the two features are inconsistent, it is necessary to normalize the lane-changing feature values. We use the min-max normalization method to make the values range from 0 to 1. If x is a set of feature values, the normalized x' can be calculated as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (1)$$

Step 2: *K*-means clustering of trajectory points. The sample set $S_p = \{S_{p1}, S_{p2}, \dots, S_{pn}\}$ is established according to the lane-changing features of trajectory points, where n is the number of trajectory points. The sample point S_{pi} corresponds to the two normalized lane-changing eigenvalues v_1 and v_2 of the i -th trajectory point, namely $S_{pi} = (v_1, v_2)$, $i = (1, 2, \dots, n)$. Firstly, the two samples closest to the origin and the farthest samples in S_p are selected as the initial cluster centers. Then, calculate the distance between each sample and the two cluster centers, and assign each sample to the closest cluster. Each cluster center and the samples assigned to it represent a cluster. Once all samples have been allocated, the cluster centers are recalculated based on the samples contained in each cluster. The sample allocation and cluster center calculation process are repeated until no samples are reassigned to different clusters. Then the category to which each sample belongs is the final clustering result. The trajectory points corresponding to the samples are

divided into two categories according to the final clustering results of the samples.

Step 3: Determination of cluster category. After classifying trajectory points, the meaning of the point cluster category is further determined according to the location of the cluster center. Compared with nonlane-changing trajectory points, lane-changing trajectory points have larger eigenvalues. Therefore, the trajectory points whose cluster center is further from the coordinate origin are regarded as lane-changing, and the other points are regarded as nonlane-changing.

3.2. Construction of Lane-Level Map. The construction of lane-level map is explained in the following sections.

3.2.1. Trajectory Distribution Modeling on the Road Cross Section. Lane centerlines can be extracted using a constrained Gaussian mixture model (CGMM) [13]. The basic principle is to analyze the density distribution of trajectory data on a road cross section. Constrained by the prior knowledge of lane distribution law, the trajectory density distribution is fitted on the premise of knowing the number of lanes. A lane's centerline position and width can be determined by the fitted trajectory density distribution on a road cross section. Furthermore, the lane centerlines are obtained by the CGMM for multiple consecutive road cross sections.

The CGMM method ignores the difference in positioning errors of vehicle trajectories. Tang et al. [15] found that some trajectories have high positioning accuracy while

others have low accuracy on the same road. Therefore, we improve the original CGMM and propose a weighted constrained Gaussian mixture model (WCGMM). The weighted model considers the difference in trajectory positioning accuracy. Firstly, it estimates the positioning accuracy of trajectories. Then, the accuracy is taken as a weight in the original constrained Gaussian mixture model.

Vehicles usually travel steadily along the extended direction of a road. Thus, the trajectory of a vehicle will be a smooth curve similar to the road's centerline. When the positioning accuracy is low, the position deviation of points on the trajectory is significant. It results in a large difference between the trajectory shape and the shape of the road centerline. Therefore, we compare the similarity between the vehicle trajectory and the road centerline to estimate the positioning accuracy of trajectories. The similarity can be measured by the standard deviation of the distance between all points and the road centerline. When a trajectory's shape is the same as the road centerline, the standard deviation of the distance between all the trajectory points and the road centerline is equal to 0. If the trajectory shape differs significantly from the road centerline, the standard deviation of the distance between all trajectory points and the road centerline is also significant. Figure 4 visualizes the results of the standard deviation of the distance between the points and the road centerline. In this figure, the red trajectories have a more significant standard deviation, and the green trajectories have a minor standard deviation. It can be found that the positioning accuracy of a trajectory is positively correlated with the standard deviation of the distance between all trajectory points and the corresponding road centerline.

Suppose the distances between the trajectory points and the corresponding road centerline are d_1, d_2, \dots, d_n (n is the number of trajectory points), and the average of these distances is \bar{d} . Then, the positioning accuracy ω of the trajectory can be calculated by equation (2). After estimating the positioning accuracy of each trajectory, a weighted constrained Gaussian mixture model is constructed based on the trajectories for each road cross section.

$$\omega = \sqrt{\frac{n}{\sum_{i=1}^n (d_i - \bar{d})^2}}. \quad (2)$$

Before discussing the WCGMM method, we first introduce the basic concepts of the Gaussian mixture model (GMM). The GMM can be formulated by mixing multiple single Gaussian distributions. Every Gaussian distribution can be regarded as a component of the GMM. Therefore, the probability density distribution $p(x|\Theta)$ of the GMM can be expressed as

$$p(x|\Theta) = \sum_{j=1}^K \varphi_j \Psi(x|\mu_j, \sigma_j), \quad (3)$$

where $\Theta = \{\varphi_1, \dots, \varphi_K, \theta_1, \dots, \theta_K\}$ are parameters of the GMM. $\varphi_1, \dots, \varphi_K$ is the probability that the sample value x belongs to each component. The sum of all the probabilities is equal to 1. $\theta_j = \{\mu_j, \sigma_j\}$ is the parameter of the j -th Gaussian



FIGURE 4: Visualization of the standard deviation of the distance between trajectory points and the road centerline.

component. μ_j is the mean, and σ_j is the standard deviation. K is the number of Gaussian components in the GMM. $\Psi(x|\mu_j, \sigma_j)$ is the probability density function of a single Gaussian distribution, and $\Psi(x|\mu_j, \sigma_j)$ can be expressed as

$$\Psi(x|\mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x - \mu_j)^2}{2\sigma_j^2}\right). \quad (4)$$

Gebru et al. [25] further introduced the concept of sample weights based on the GMM. It combines the sample weights and the GMM by treating the weight ω of sample x as ω times equivalent observations of x . Then, the probability density distribution $p(x|\Theta, \omega)$ of the weighted GMM can be derived as

$$p(x|\Theta, \omega) = \sum_{j=1}^K \varphi_j \Psi\left(x|\mu_j, \frac{\sigma_j}{\omega}\right). \quad (5)$$

The weighted GMM can model the density distribution of vehicle trajectories on road cross sections. Sample value x can be expressed as the position of a trajectory on the road cross section. ω is the estimated positioning accuracy of each trajectory segment. K can be expressed as the number of lanes. φ_j is equivalent to the ratio of lane traffic flow to roadway traffic flow. μ_j corresponds to the position of each lane centerline. σ_j is the dispersion of trajectories on each lane. σ_j is related to the width of a lane. According to urban road construction standards, the width of each lane on the same roadway should be equal. Thus, σ_j of every Gaussian distribution is the same in a weighted GMM, that is, $\sigma_1 = \sigma_2 = \dots = \sigma_K = \sigma$. We call the weighted GMM constrained by the same σ as the weighted constrained GMM (WCGMM). Lane-level road information can be obtained by solving the parameters of the model.

The EM algorithm calculates the parameters of the WCGMM. The EM algorithm is widely used to solve the latent variable model. It can estimate the maximum likelihood probability, or posterior probability, of a latent variable model in an iterative manner. The maximum posterior probability method is used to estimate the parameters. To express model parameters in the iterative process of the EM algorithm, we represent model parameters Θ as $(\varphi_j(m), \mu_j(m), \sigma(m))$, where m is the solution of the m -th iteration process. For the initial value of parameters in the first iteration, we set them as $\varphi_1(0) = \varphi_2(0) = \dots = \varphi_K(0) = 1/K$, and the setting of $\sigma(0)$ can refer to the standard width of lanes.

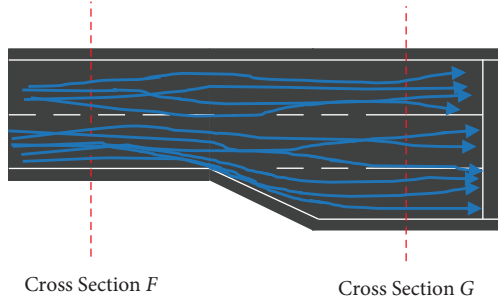


FIGURE 5: The relationship between lane-changing behavior and the lane counts.

We set $\sigma(0)$ to 1.75 meters and set the initially estimated position $\mu_j(0)$ by $\sigma(0)$ and lanes number K .

The above EM algorithm is divided into two steps: step E estimates the probability of each sample belonging to each Gaussian component. Step M updates the model parameters. We alternate the step E and step M iteratively to make the model gradually approach the maximum posterior probability. Among them, the calculation method of step E is

$$\gamma_{ij}^{(m)} = \frac{\varphi_j^{(m)} \Psi(x_i | \mu_j^{(m)}, \sigma / \omega_i^{(m)})}{\sum_{l=1}^K \varphi_l^{(m)} \Psi(x_i | \mu_l^{(m)}, \sigma / \omega_i^{(m)})}, \quad (6)$$

where x_i is the i -th sample value, ω_i is the weight of x_i , and $\gamma_{ij}(m)$ represents the probability that x_i belongs to the j -th Gaussian component during the m -th iteration. Step M updates the parameters of the WCGMM according to the probabilities that each sample calculated in step E belongs to different Gaussian components. If the total number of samples is n , the calculation method is

$$\begin{aligned} \varphi_j^{(m+1)} &= \frac{\sum_{i=1}^n \gamma_{ij}^{(m)}}{n}, \\ \mu_j^{(m+1)} &= \frac{1}{\sum_{i=1}^n \omega_i \gamma_{ij}^{(m)}} \sum_{i=1}^n \omega_i \gamma_{ij}^{(m)} x_i, \\ \sigma^{(m)} &= \sqrt{\frac{1}{\sum_{i=1}^n \gamma_{ij}^{(m)}} \sum_{i=1}^n \omega_i \gamma_{ij}^{(m)} (x_i - \mu_j^{(m+1)})^2}. \end{aligned} \quad (7)$$

When the number of lanes on a road cross section is known, the value of each unknown parameter can be calculated through the EM algorithm. Then the lane centerline and lane width can be estimated. Meanwhile, this model's likelihood can be calculated using model parameters. The likelihood can be understood as the probability of the current trajectory state distribution under the conditions of the known number of lanes and model parameters. It can be expressed as equation (8). In the next section, we will use the log-likelihood of the model to extract the number of lanes.

$$L(X|\Theta, K) = \prod_{i=1}^n p(x_i | \Theta, \omega_i, K). \quad (8)$$

3.2.2. Extraction of Lane Counts on Continuous Road Cross Sections. The ratio of vehicle lane-changing behavior between adjacent road cross sections is related to the number of lanes. If the number of lanes increases or decreases, many vehicles will change lanes. When the traffic volume of each lane is approximately the same, the changes in lane counts can be estimated by the ratio of lane-changing vehicles to all vehicles on the road segment. As shown in Figure 5, when the number of lanes changes from two to three, two of the three lanes on road cross section G is the continuation of the two lanes on road cross section F. The remaining one is the newly added lane. Vehicles on the newly-added lane must go through a lane change to reach the newly-added lane. Therefore, the number of lanes changes from F to G can be estimated by counting the ratio of vehicles that have caused lane-changing behavior.

Under the premise that trajectory distribution on the road and the number of lane-changing vehicles are known, the lane counts estimation can be modeled by the hidden Markov model (HMM). HMM is a Markov process with hidden states and observations [26]. Each state has an observation probability for the possible observations. The transition probability defines the state-to-state transition. The hidden state sequence can be generated by maximizing the overall probability given a series of observations. HMM mainly includes five elements, namely two state sets and three probability matrices [27]:

Hidden state H: the hidden state in HMM cannot be obtained by direct observation. These states satisfy the Markov property. Each hidden state is only related to its neighboring hidden states.

Observation state O: A state that can be obtained by direct observation and is related to an implicit state in HMM.

State transition matrix B : the matrix describes the transition probability between the hidden states in HMM. $B = [b_{ij}]_{M \times M}$, where $b_{ij} = P(h_{t+1} = H_j | h_t = H_i)$, $1 \leq i, j \leq M$, and M is the number of hidden states. b_{ij} represents the probability that the hidden state is H_j at $t + 1$, and the hidden state is H_i at t .

Observation state matrix C : this matrix describes the probability of the current observation state under the condition that the hidden state is known. $C = [c_j(k)]_{M \times N}$, where $c_j(k) = P(o_t = O_k | h_t = H_j)$, $1 \leq k \leq N$, N is the number of observation states. $c_j(k)$ represents the probability that the observed state is O_k , and the hidden state is H_j at time t .

The initial state matrix η : $\eta = (\eta_i)$, $\eta_i = P(h_1 = H_i)$, $1 \leq i \leq N$. It represents the probability matrix of the hidden state at the initial time $t = 1$.

The number of lanes at multiple road cross sections can be considered a Markov process. It means the number of lanes at each road cross section is only related to adjacent road cross sections. Therefore, the HMM model can be established by taking the number of lanes on multiple consecutive road cross sections as the hidden state, assuming

that the number of lanes at multiple consecutive road cross sections is $H = \{h_1, h_2, \dots, h_T\}$, where T is the number of road cross sections. The trajectory distribution can model the observation state O on each road cross section, namely $O = \{o_1, o_2, \dots, o_T\}$. According to the fitting result of the WCGMM on the trajectory distribution, the observation state matrix C can be calculated. Each element $c_j(k)$ in C can be expressed as

$$c_j(k) = \ln(L(O_k | \Theta, H_j)), \quad (9)$$

where $L(O_k | \Theta, H_j)$ is the likelihood of the WCGMM under the condition that the number of lanes is H_j . It can be calculated by equation (8). The value of this probability is usually minimal because the probability is in the form of multiplication of multiple probabilities. Therefore, we use the log-likelihood to calculate the observation state. The maximum log-likelihood of all observation states is used for the min-max normalization to facilitate computer processing.

The state transition matrix B can be defined according to the ratio of the lane-changing vehicle. Each element b_{ij} in B can be expressed as

$$b_{ij} = P(h_{t+1} = H_j | h_t = H_i) = \exp\left(-\left(|H_i - H_j| - \beta_{t,t+1} \cdot \max(H_i, H_j)\right)^2\right), \quad (10)$$

where $\beta_{t,t+1}$ is the ratio of lane-changing vehicles between the t and $t+1$ road cross sections. The product of $\beta_{t,t+1}$, and the maximum number of lanes on the adjacent road cross section is the number of changing lanes. According to the difference between hidden states H_i and H_j , the number of changing lanes can also be obtained. Therefore, the probability of lane counts on the current and previous road cross sections is H_j and H_i , which can be calculated according to the number of changing lanes. Taking into account that some vehicles may change lanes due to the need to overtake or turn even if there is no new lane, $\beta_{t,t+1}$ is expressed as

$$\beta_{t,t+1} = \frac{|CG_{\text{left}} - CG_{\text{right}}|}{CR_{\text{total}}}, \quad (11)$$

where CG_{left} and CG_{right} represent the number of vehicles changing lanes to the left and right, respectively. CR_{total} represents the number of all vehicles between two adjacent road cross sections. $\beta_{t,t+1}$ is calculated this way because the probability of vehicles changing to the left and right sides is considered equal when no additional lanes exist. Using the difference in the number of vehicles changing lanes to the left and right can avoid the influence of overtaking or turning.

The number of lane-changing vehicles can be extracted according to the lane-change trajectory points identified in Section 3.1. First, we count all lane-change trajectory points between two adjacent road cross sections. Then these points are classified into left and right categories according to the change in direction. Finally, the number of vehicles changing

direction to the left or right is counted by the vehicle's ID of trajectory points.

The observed state probability can directly represent the initial state matrix. We take the maximum number of lanes as a parameter of the algorithm, and the candidate set of hidden state variables can be determined according to the maximum number of lanes. The solution to the hidden state in HMM is to find the number of lanes on each road cross section with the most significant overall probability. Generally, it can be solved by the Viterbi algorithm [28]. After using the Viterbi algorithm to find the number of lanes in each road cross section, the WCGMM corresponding to each road cross section can be determined. Then the lane information on a road cross section can be obtained using model parameters.

3.2.3. Extraction of Lane Centerlines. The HMM and the WCGMM can generate lane centerline nodes on road cross sections. Then, we need to construct lane centerlines by connecting these nodes on multiple consecutive road cross sections. To make the constructed lane centerlines as close to the actual lane centerlines as possible, we need to divide the road into as many cross sections as possible. These multiple road cross sections are set along the centerline of a road at a fixed length interval to extract the lane centerline nodes. We take the fixed-length interval of road cross sections as a parameter of the algorithm. Then, we can extract multiple-lane centerline nodes with more than two lanes on one road cross section. Besides, the number of centerline nodes on adjacent road cross sections may differ when lanes change. It is necessary to determine which two nodes on the adjacent road cross sections need to be connected. A method for connecting lane centerline nodes is proposed based on the minimum matching distance, which includes the following steps:

Step 1: Index the lane centerline nodes. Sort the nodes on a cross section according to the distance between the node and the road's centerline. Then number the nodes according to the sort order. As shown in Figure 6, for the nodes on road cross section G , the closest node to the centerline is G_1 , followed by G_2 and G_3 , until the n -th node is G_n .

Step 2: Build a candidate match set. Since the number of lanes on two adjacent road cross sections may differ, we use the road cross section with fewer lanes as a reference. Then we match the nodes on the cross section with more lanes on the reference road cross section. If two adjacent road cross sections are F and G , then the number of lanes in F is m , the lane counts in G is n , and $m < n$. Therefore, the set of nodes in F is $F_{\text{Node}} = \{F_1, F_2, \dots, F_m\}$, and the set of nodes in G is $G_{\text{Node}} = \{G_1, G_2, \dots, G_n\}$. According to the node index in G , the candidate matching set for F_{Node} is constructed. Each candidate matching set for F_{Node} is $\{G_i, G_{i+1}, \dots, G_{i+m-1}\}$, with i from 1 to $n+1-m$. According to each candidate matching set, a matching pair can be constructed, that

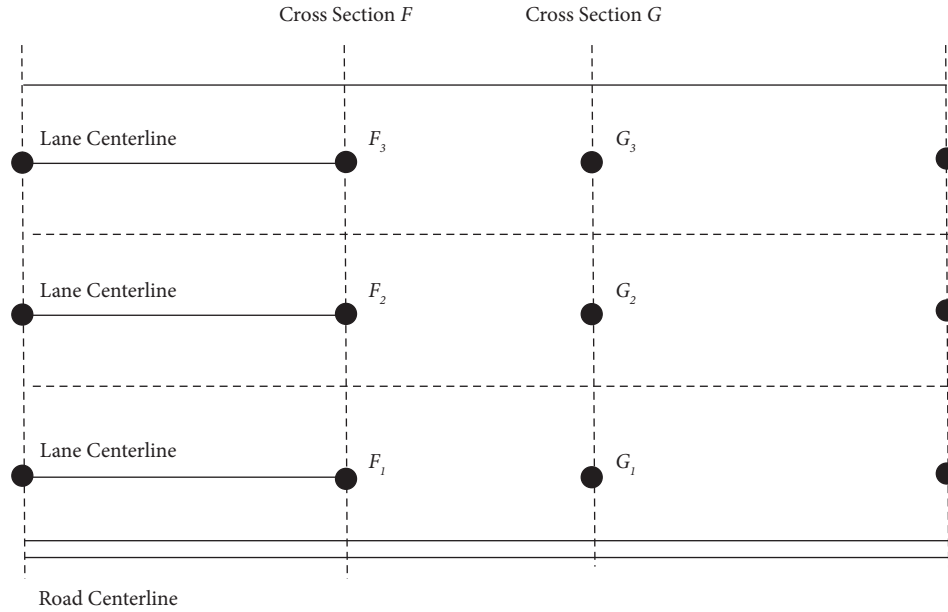


FIGURE 6: Lane centerline node index.

is, F_1 matches G_i , F_2 matches G_{i+1} , and F_m matches G_{i+m-1} .

Step 3: Calculate the matching distance. For each candidate matching set, we calculate the distance of each matching pair in the matching set. Then we calculate the sum of all distances for each matching pair. The sum is the matching distance of the whole candidate matching set.

Step 4: Construct the centerlines of lanes. The candidate matching set with the smallest matching distance is selected. We connect the corresponding nodes according to each matching pair, as shown by the solid red line in Figure 7. For the unmatched node G_u on road cross section G , we find the closest matching node G_v on G . Then we connect G_u to the corresponding matching node of G_v on road cross section F , as shown by the red dashed line in Figure 7.

4. Experiments

The experiments are described in the following sections.

4.1. Experimental Data Collection and Preprocessing. Mapillary is a street-level image data-sharing platform based on geospatial tags. Users can upload vehicle trajectory data with geographic location tags for individuals or teams. Through the service interface and development toolkit provided by Mapillary, users can download crowdsourced vehicle trajectory data on the platform. We found that the number of trajectories uploaded by users in the San Francisco area is relatively large, and the higher the road grade, the higher the trajectory coverage rate. In order to ensure that there are enough trajectories to cover the road area, we collect vehicle trajectories on motorway sections, including

US 101, CA 1, I280, and I80. Figure 8 shows the result of superimposing collected trajectory data on Google Satellite Maps. As can be seen from Figure 8, the research area contains complex road scenes such as curves, tunnels, and overpasses. The blue lines in Figure 8 are trajectory data. It contains 3,728 trajectories and 557,924 trajectory points. These trajectories are sampled at intervals ranging from 1 to 10 seconds. The average sampling interval is 2.36 seconds. The positioning errors of some trajectories exceed 100 meters. We treat these vehicle trajectories with significant positioning errors as noise data. Therefore, this data needs to be preprocessed before lane information extraction to eliminate the noise.

We download the road centerline data within the research area from the OSM website. The road centerlines are used to estimate the positioning error of each trajectory using the error calculation method proposed in Section 3.2. We introduced the natural discontinuity method [29] to determine the positioning error threshold of noise. Trajectories with errors higher than the threshold are regarded as noisy. Figure 9 shows the vehicle trajectories after preprocessing on one of the road sections in the research area. Trajectories on the left and right sides of the road are distinguished. We use vehicle trajectories on each side to extract lane information, respectively. Table 1 shows the parameter setting of the proposed method. According to road construction standards, the maximum number of lanes on urban roads is set at 6. Therefore, the candidate hidden value in the HMM model is $\{1, 2, 3, 4, 5, 6\}$. The length of the interval between road cross sections is related to the algorithm efficiency and the smoothness of the extracted lane centerlines. The longer the interval, the faster the calculation speed, while the shorter the interval, the smoother the extracted lane centerline. We set the length interval to 20 meters considering the above two factors.

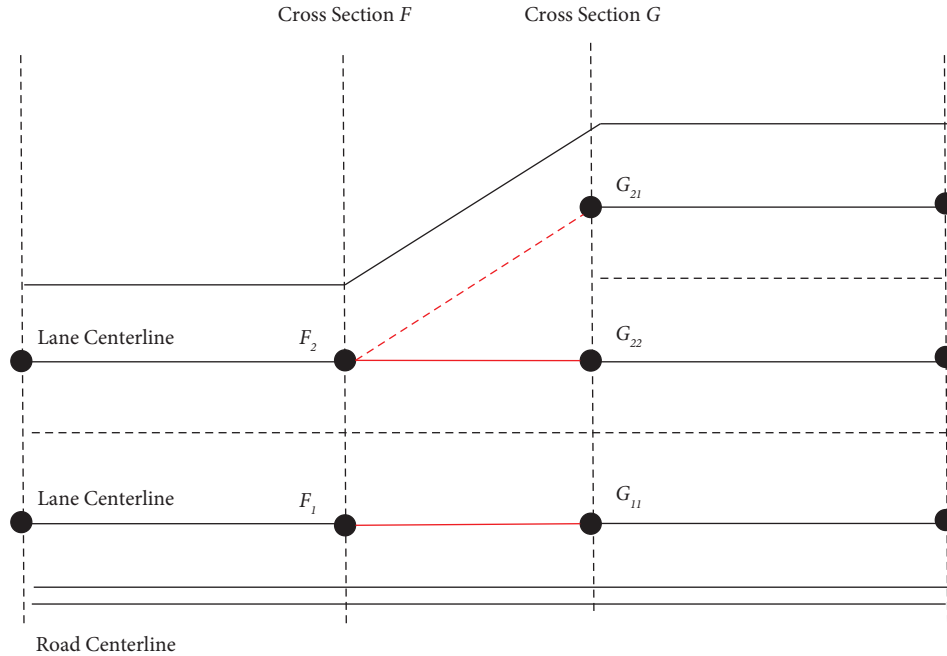


FIGURE 7: Lane centerline node connection.



FIGURE 8: Vehicle trajectory data in the study area.



FIGURE 9: Preprocessed vehicle trajectory data.

TABLE 1: Parameters setting of the proposed method.

Parameter	Value
Max number of the lanes	6
Length interval of road cross sections	20 meters

4.2. Extraction of Lane-Changing Trajectory Points. The extraction of lane-changing trajectory points is the first step in the proposed method. By observing the trajectory distribution on different roads, we found a specific correlation between the lane-changing behavior of vehicles and the change in the number of lanes. In order to verify the

rationality of this assumption, we selected two road sections (road sections A and B in Figure 10) in the study area to carry out experiments. The number of lanes on road section A remains the same, and the number of lanes on road section B changes from 4 to 5. The traffic volumes of the two road sections are similar.

The purple and pink points in Figure 10 are the vehicle lane-changing trajectory points extracted by the method proposed in this paper. It can be seen that more vehicles have changed lanes on road B. The detection results were consistent with the visual results. We further count the number of detected lane-changing vehicles. As shown in Figure 11, it can be seen that more vehicles change lanes to the right in road section B. This is because some vehicles need to enter the new lane on the right. There are also more vehicles changing lanes to the right than to the left on road A. We think this may be caused by the driving habits of drivers. The driving habits of drivers can be further studied quantitatively. However, it is not significant compared with the difference caused by adding lanes. Thus, we did not take this factor into account in our method.

4.3. Construction of Lane Centerlines. Accurately fitting the trajectory distribution on the road cross section is the key to extracting the lane centerline. Based on the existing CGMM, we propose a weighted constrained Gaussian mixture model combined with the trajectory positioning error and verify the method's effectiveness through the trajectory data on road A in Figure 10. Figure 12 shows the trajectory distribution fitting results of the WCGMM method under the assumption of different numbers of lanes. Unlike the original Gaussian mixture model, the weighted Gaussian mixture model is related to the error of each sample. Thus, the fitted distribution of the Gaussian mixture model is not a



FIGURE 10: Extraction results of lane-changing trajectory points.

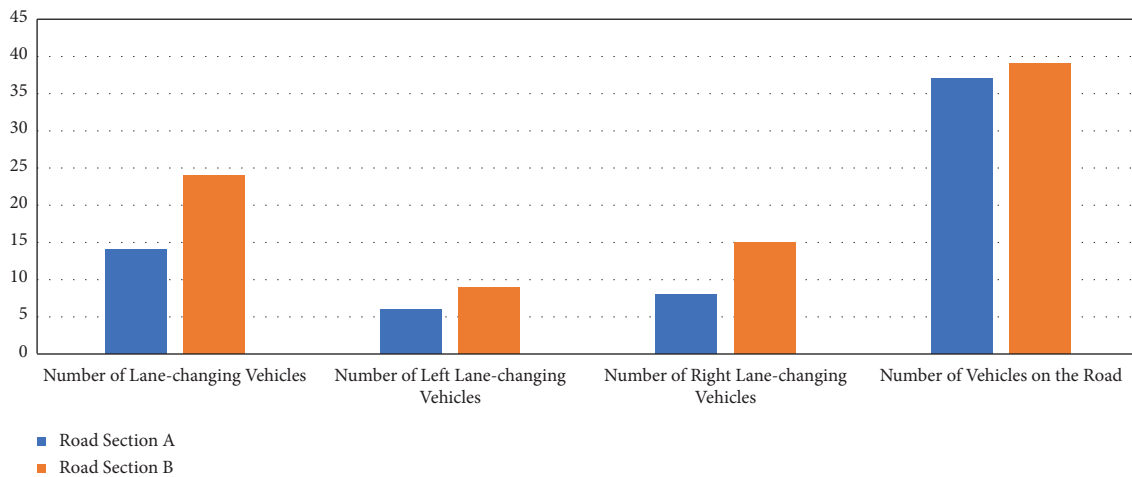


FIGURE 11: Statistical results of lane-changing vehicles.

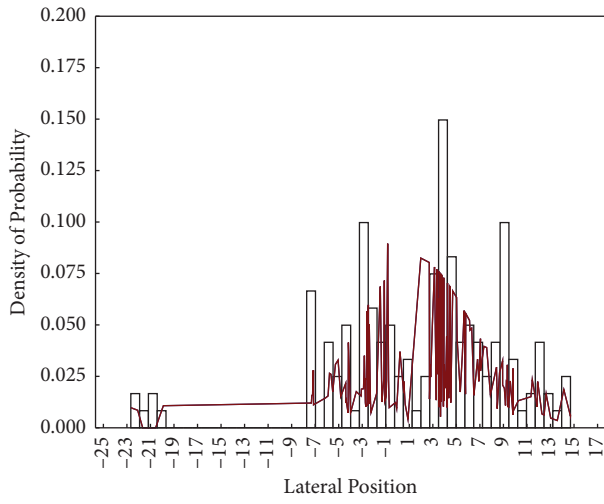
smooth curve. Figure 13 is the log-likelihood for each subgraph in Figure 12. It can be seen that the largest log-likelihood is under the condition of 4 lanes, consistent with the results observed in Figure 10.

An accurate estimation of the number of lanes is the basis of lane centerline extraction. In order to accurately estimate the number of lanes from the trajectory data, our method has made two improvements. First, a weighted-based constrained Gaussian mixture model is proposed. Based on the original constrained Gaussian mixture model, the influence of trajectory positioning errors is considered, and the weight of high-precision trajectory data is increased while the impact of low-precision trajectory data errors is reduced on model fitting accuracy. Then, a hidden Markov model is used to model the correlation between vehicle lane-changing behavior and lane number changes to improve the consistency of lane number estimates on adjacent road cross sections.

Figure 14 shows the number of lanes extracted using the trajectory data in Figure 9. The actual lane number of the road is obtained by manually interpreting the Google Satellite Map at each road cross section. Figure 14(a) is an improved method that only adds a weighted constrained Gaussian mixture model. Figure 14(b) is the extraction result

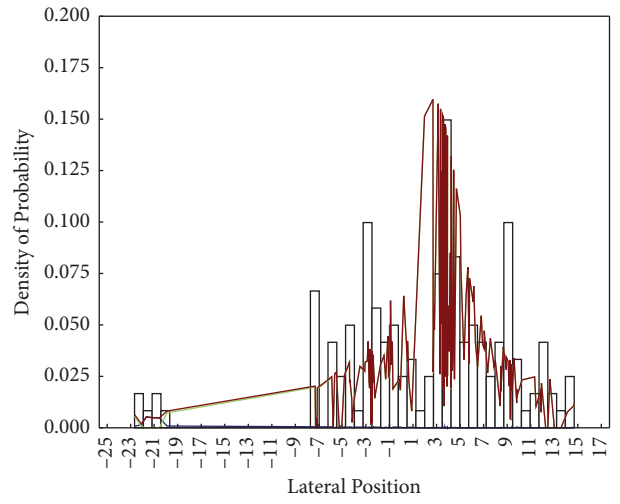
of the number of lanes obtained using a weighted constrained Gaussian mixture model and a hidden Markov model to fit the distribution of road transverse and longitudinal trajectories. It can be found that the continuity of the number of lanes on the road is more robust when both improvement methods are used at the same time, compared with only using the weighted constrained Gaussian mixture model.

In order to further verify the effectiveness of the method, we use all the trajectory data collected in the study area to construct a lane-level road map, as shown in Figure 15. In order to show the detailed information extracted from the lane centerline more clearly, we selected three representative areas, a, b, and c, in Figure 15. Area a is the scene where vehicles pass through the tunnel; area b is a curve; and area c is a viaduct where multiple roads converge. Figure 16 is the result of lane centerline extraction in three scenarios. Figure 16(a) superimposes the extracted lane centerlines and experimental trajectory data, and Figure 16(b) superimposes the extracted lane centerlines and reference lane centerlines. The centerline of the reference lane is manually compared with Google Maps' vectorization. It should be noted that due to the positioning error of Google Maps itself, there will also be errors in the centerline of the reference lane generated by



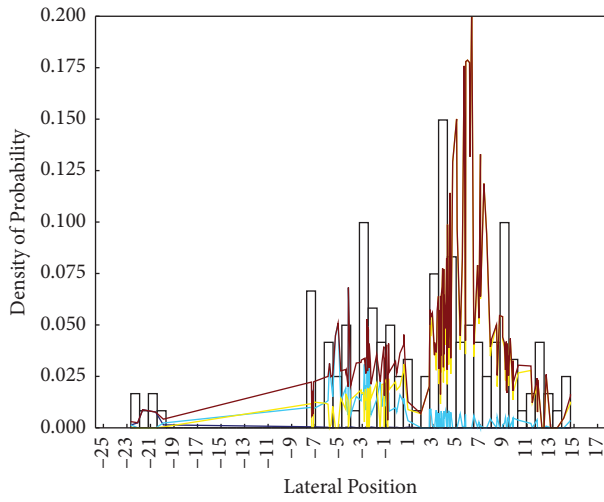
□ Histogram of Trajectory Distribution
— Density of Mixture Distribution Probability

(a)



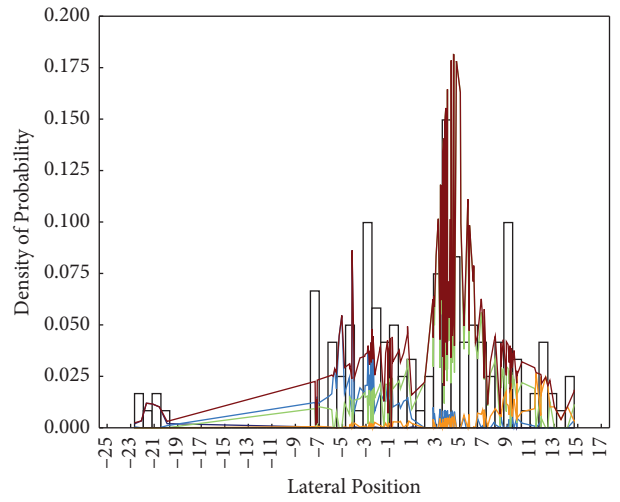
□ Histogram of Trajectory Distribution
— Density of Mixture Distribution Probability

(b)



□ Histogram of Trajectory Distribution
— Density of Mixture Distribution Probability

(c)



□ Histogram of Trajectory Distribution
— Density of Mixture Distribution Probability

(d)

FIGURE 12: Continued.

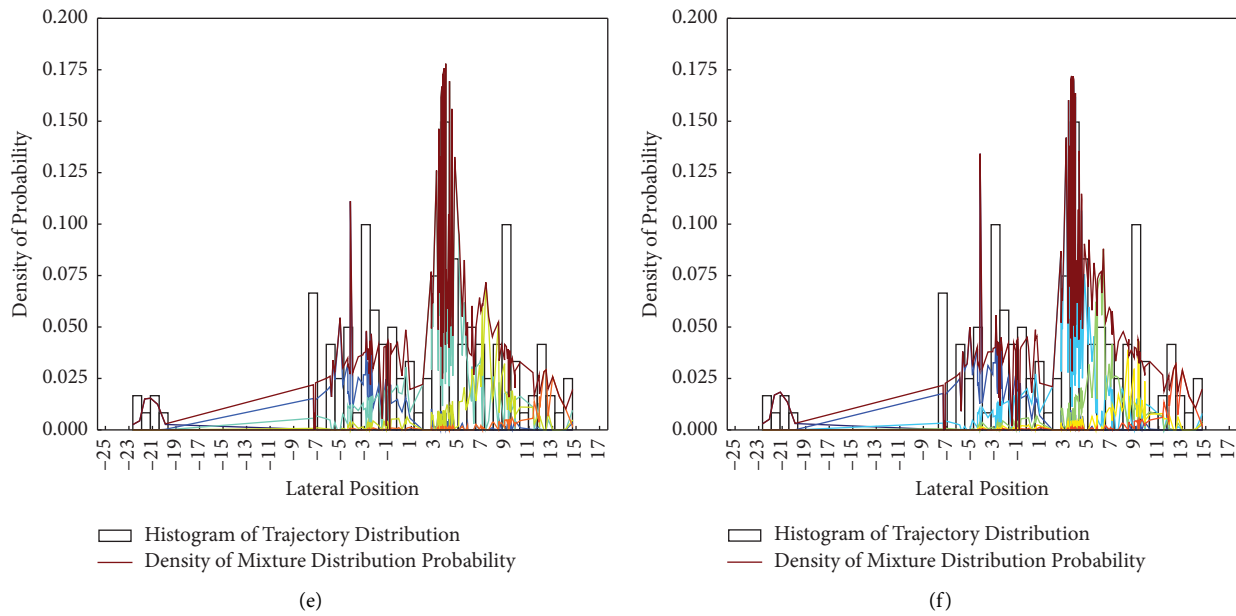


FIGURE 12: Fitting results of trajectory distribution on road cross-section based on WCGMM: (a) condition of one lane, (b) condition of two lanes, (c) condition of three lanes, (d) condition of four lanes, (e) condition of five lanes, and (f) condition of six lanes.

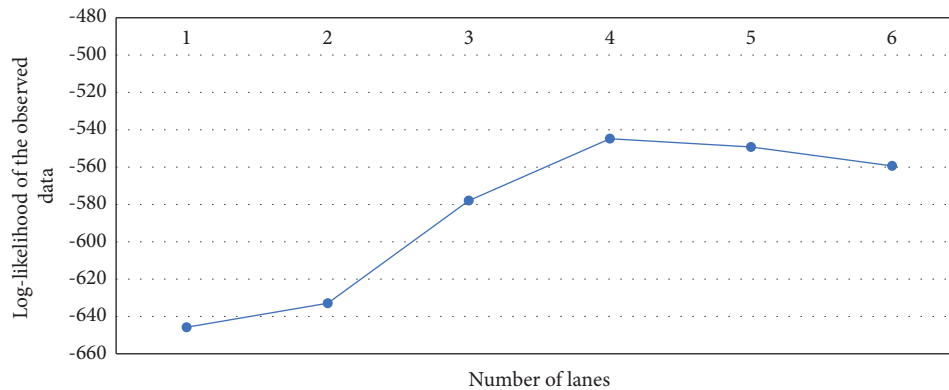


FIGURE 13: The log-likelihood of trajectory distribution under different numbers of lanes.

the manual vectorization of Google Maps. We consider the reference lane centerline error acceptable compared to the lane centerline extracted by trajectory. Because many lane-level road maps also use Google Maps as the primary data source.

Figure 16 shows that the lane centerline extraction results in area b are relatively more accurate, the lane width extracted in area a is wider than the actual width, and the number of lanes and lane centerline positions of some roads in area c are incorrectly extracted. Comparing the extraction results of the three regions, we found that two reasons may reduce the extraction accuracy of lane information. The first reason is the trajectory error. Although the WCGMM model considers the influence of the trajectory error, it cannot eliminate the trajectory positioning error. When the error is too large and the number of samples is insufficient, even if the number of lanes can be estimated correctly, there will still be a significant deviation in the position of the fitted

centerline. The second reason is insufficient trajectory coverage. When the trajectory distribution on a specific lane is sparse or even has no trajectory, the uncertainty of the extracted lane information will increase. In addition, this paper's method only applies to the extraction of lane information from road segments. For the lane information of road intersections, the steering behavior of different vehicle trajectories is used to further extract the routing information of the intersection after extracting the lane information on road segments.

4.4. Quantitative Evaluation. Existing methods for extracting lane information using vehicle trajectories mainly include KDE (kernel density estimation), proposed by Uduwaragoda et al. [12], and CGMM (constrained Gaussian mixture model) proposed by Chen and Krumm [13]. Among them, the CGMM method uses constraint conditions such as

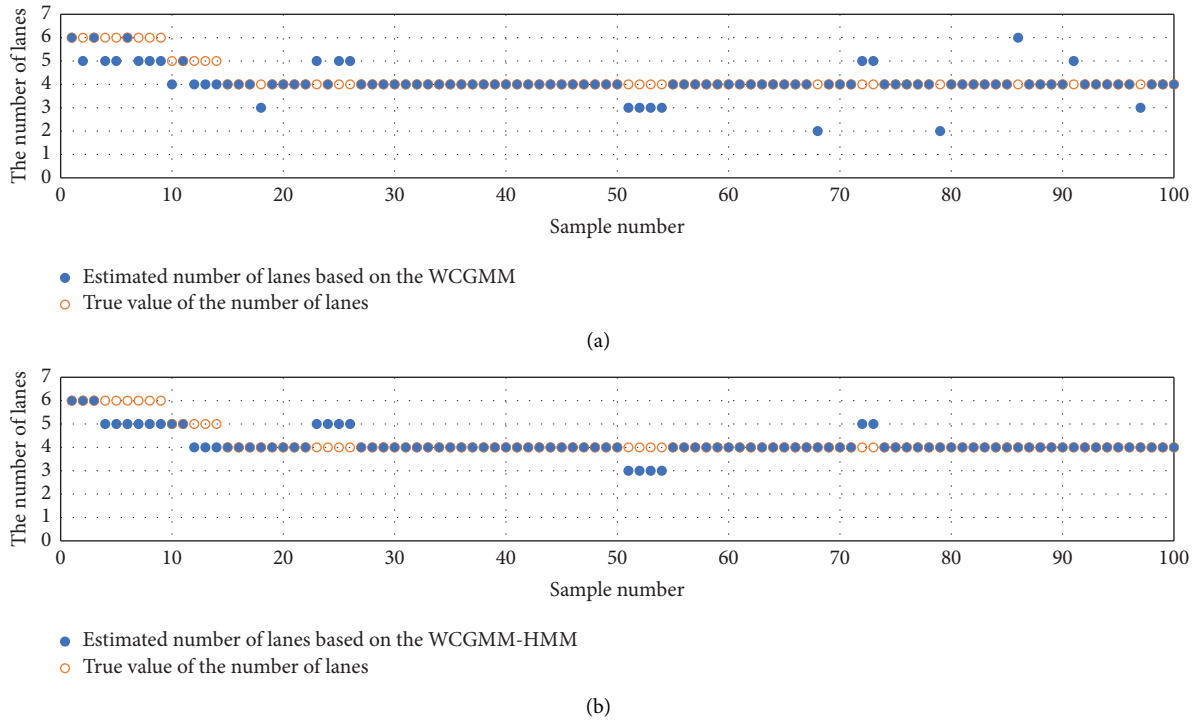


FIGURE 14: The number of actual lanes and the number of extracted lanes on multiple consecutive road cross sections: (a) estimated results based on the WCGMM; (b) estimated results based on the WCGMM and HMM.



FIGURE 15: Lane-level road map generation results based on the proposed method.

trajectory coverage width and lane width to fit a Gaussian mixture distribution on the trajectories of a road cross section. Then, it extracts lane information according to the parameters of the fitted Gaussian mixture model. The KDE method estimates the trajectory density distribution of vehicle trajectories in the X -axis and Y -axis through kernel density. Then it determines the position of lane centerlines according to the peak points of the trajectory density distribution. The proposed method combines the

constrained Gaussian mixture model with lane-changing features of trajectories. The method uses a weighted constrained Gaussian mixture model and a hidden Markov model to estimate the number of lanes and extract lane centerlines.

The results of the lane count extraction experiment with different methods are shown in Table 2. They are obtained by calculating the ratio of correctly extracted road cross sections to total road cross sections. These results show that

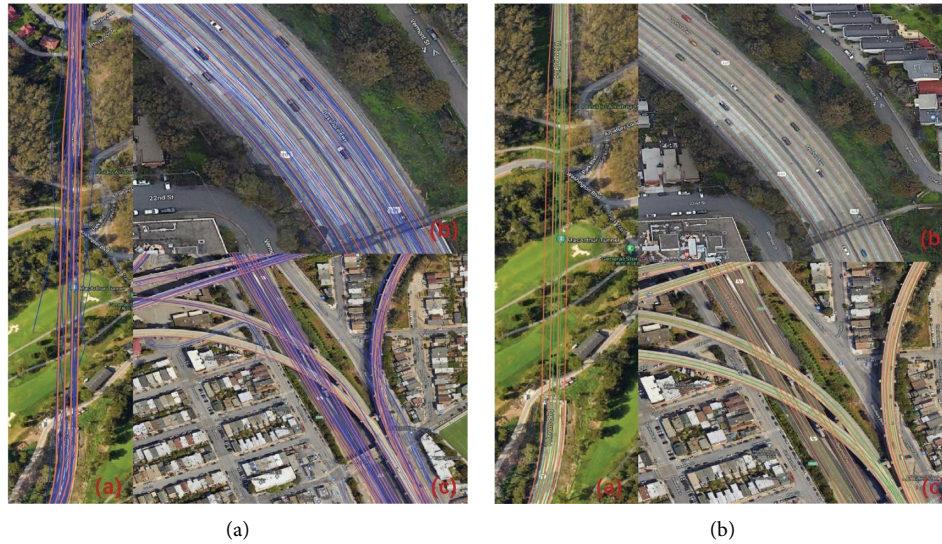


FIGURE 16: Lane-level road map generation results based on the proposed method within the three selected areas: (a) the extracted lane centerlines and experimental trajectory data; (b) the extracted lane centerlines and reference lane centerlines.

TABLE 2: Accuracy comparison of lane counts extraction

Lane counts extraction methods	Accuracy (%)
KDE [12]	65.3
CGMM [13]	75.5
Proposed WCGMM	76.8
Proposed WCGMM and HMM	78.6

TABLE 3: Comparative analysis of the position accuracy of lane centerlines.

	CGMM	KDE	Proposed WCGMM	Proposed WCGMM and HMM
Average (m)	4.33	6.52	3.82	3.15
Minimum (m)	0.01	0.17	0.05	0.01
Maximum (m)	8.25	12.11	9.54	8.32
Standard deviation (m)	0.94	2.57	0.89	0.55

the accuracy of lane count recognition based on GMM is better than kernel density estimation. Compared with CGMM and KDE, the proposed method uses trajectory information between adjacent road cross sections and lane-changing features. It can improve the accuracy of lane count extraction. In addition, compared with only using the WCGMM model, combining HMM and WCGMM can further improve the algorithm's accuracy.

The comparative analysis of the lane centerline extraction experiment needs to use the actual lane centerline as a reference. Google satellite map is used as a reference. The reference lane centerlines are obtained by manually vectorizing lane centerlines in the research area. The extraction error can be estimated according to the distance between the extracted and reference centerline nodes. The average, minimum, maximum, and standard deviation of the distance between each node are used as the index of extraction accuracy. Table 3 shows the evaluation results of each method. It can be found that the kernel density estimation method has lower accuracy than GMM because the former needs to select an appropriate kernel density radius. If the

trajectory positioning accuracy is low, it is not easy to estimate the accurate density distribution of vehicle trajectories on the road. The Gaussian mixture model can achieve more accurate results by adding the constraints of prior knowledge. The proposed method takes trajectory positioning accuracy as the weight in the constrained Gaussian mixture model. It can further improve the accuracy of lane centerline extraction. Thus, the average error of the proposed method is minor. Using both the WCGMM and the HMM model can reduce the error variance compared to only using the WCGMM model. It is because the HMM uses the vehicle's lane-changing behavior to model the longitudinal trajectory distribution of the road so that the lane information on different road cross sections has a better consistency.

5. Conclusions

This article studies the problem of using crowdsourced vehicle trajectories to extract lane centerlines. A high-precision lane centerline extraction method is proposed

based on the weighted, constrained Gaussian mixture model and the hidden Markov model. The proposed method first introduces the Frenet coordinate system based on road centerlines. Then it fits the vehicle trajectory in the Frenet coordinate system. Based on the fitted trajectory, it extracts critical features that best distinguish between lane-changing and nonlane-changing points. Then, vehicle trajectories with positioning errors are used to model the weighted constrained Gaussian mixture model. The hidden Markov model is constructed according to the lane-changing trajectory points to estimate the number of lanes in the WCGMM. Finally, lane centerline nodes are extracted from the parameters of the WCGMM. Lane centerlines are constructed from these nodes. Vehicle trajectories in San Francisco collected from Mapillary are used to verify and analyze the proposed method. Experimental results show that our method's lane count extraction accuracy reaches 78.6% and the average positioning error of extracted lane centerlines is 3.15 m. Compared with existing methods, the proposed method takes the trajectory positioning accuracy as a weight, which can effectively improve the accuracy of lane counts and lane centerline extraction. However, considering the practical application requirements, the proposed method still needs to improve lane centerline extraction's position accuracy and robustness. Besides, this method does not consider lane distribution at road intersections. Our future work will focus on lane information extraction in a complex road network with road intersections.

Data Availability

The data used to support the findings of this study can be obtained from the corresponding author upon request.

Additional Points

Highlights. (i) We identify the lane-changing behavior of vehicles from massive trajectories. (ii) A weighted, constrained Gaussian mixture model is proposed to describe the trajectory distribution on a road cross section. (iii) A hidden Markov model is proposed to estimate the lane counts in different road segments. (iv) A compatible and more accurate estimation of lane centerlines is achieved by considering lane-changing behavior.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Innovation and Entrepreneurship Team Import Project of Shaoguan City under Grant no. 201212176230928.

References

- [1] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.
- [2] M. Haklay and P. Weber, "Openstreetmap: user-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.
- [3] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, and L. Chen, "ALIMC: activity landmark-based indoor mapping via crowdsourcing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2774–2785, 2015.
- [4] C. Kim, S. Cho, M. Sunwoo, P. Resende, B. Bradař, and K. Jo, "Cloud update of geodetic normal distribution map based on crowd-sourcing detection against road environment changes," *Journal of Advanced Transportation*, vol. 2022, Article ID 4486177, 26 pages, 2022.
- [5] K. Van Winden, F. Biljecki, and S. Van der Spek, "Automatic update of road attributes by mining GPS tracks," *Transactions in GIS*, vol. 20, no. 5, pp. 664–683, 2016.
- [6] J. Huang, Y. Zhang, M. Deng, and Z. He, "Mining crowd-sourced trajectory and geo-tagged data for spatial-semantic road map construction," *Transactions in GIS*, vol. 26, pp. 735–754, 2021.
- [7] S. He, F. Bastani, S. Jagwani et al., "Sat2graph: road graph extraction through graph-tensor encoding," in *Proceedings of the European Conference on Computer Vision*, pp. 51–67, Springer, VisionGlasgow, UK, August 2020.
- [8] Z. Fu, L. Fan, Y. Sun, and Z. Tian, "Density adaptive approach for generating road network from GPS trajectories," *IEEE Access*, vol. 8, pp. 51388–51399, 2020.
- [9] W. C. Lee and J. Krumm, "Trajectory preprocessing," *Computing with Spatial Trajectories*, Springer, New York, NY, USA, pp. 3–33, 2011.
- [10] J. Wang, X. Rui, X. Song, X. Tan, C. Wang, and V. Raghavan, "A novel approach for generating routable road maps from vehicle GPS traces," *International Journal of Geographical Information Science*, vol. 29, no. 1, pp. 69–91, 2015.
- [11] S. Edelkamp and S. Schrödl, "Route planning and map inference with global positioning traces," *Computer Science in Perspective: Essays Dedicated to Thomas Ottmann*, vol. 2598, p. 128, 2003.
- [12] E. Uduwaragoda, A. S. Perera, and S. A. D. Dias, "Generating lane level road data from vehicle trajectories using kernel density estimation," in *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 384–391, IEEE, The Hague, Netherlands, October 2013.
- [13] Y. Chen and J. Krumm, "Probabilistic modeling of traffic lanes from GPS traces," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 81–88, San Jose, CA, USA, November 2010.
- [14] L. Tang, X. Yang, Z. Kan, and Q. Li, "Lane-level road information mining from vehicle GPS trajectories based on naïve bayesian classification," *ISPRS International Journal of Geo-Information*, vol. 4, no. 4, pp. 2660–2680, 2015.
- [15] L. Tang, X. Yang, Z. Dong, and Q. Li, "CLRIC: collecting lane-based road information via crowdsourcing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2552–2562, 2016.
- [16] M. Marinelli, G. Palmisano, V. Astarita, M. Ottomanelli, and M. Dell'Orco, "A Fuzzy set-based method to identify the car position in a road lane at intersections by smartphone GPS data," *Transportation Research Procedia*, vol. 27, pp. 444–451, 2017.

- [17] O. Roeth, D. Zaum, and C. Brenner, "Extracting lane geometry and topology information from vehicle fleet trajectories in complex urban scenarios using a reversible jump MCMC method," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, p. 51, 2017.
- [18] X. Gong, F. Wu, R. Xing, J. Du, and C. Liu, "LCBRG: a lane-level road cluster mining algorithm with bidirectional region growing," *Open Geosciences*, vol. 13, no. 1, pp. 835–850, 2021.
- [19] M. A. Arman and C. M. J. Tampère, "Road centreline and lane reconstruction from pervasive GPS tracking on motorways," *Procedia Computer Science*, vol. 170, pp. 434–441, 2020.
- [20] M. A. Arman and C. M. J. Tampère, "Lane-level routable digital map reconstruction for motorway networks using low-precision GPS data," *Transportation Research Part C: Emerging Technologies*, vol. 129, Article ID 103234, 2021.
- [21] L. Zheng, B. Li, B. Yang, H. Song, and Z. Lu, "Lane-level road network generation techniques for lane-level maps of autonomous vehicles: a survey," *Sustainability*, vol. 11, no. 16, p. 4511, 2019.
- [22] X. Yang, L. Tang, K. Stewart, Z. Dong, X. Zhang, and Q. Li, "Automatic change detection in lane-level road networks using GPS trajectories," *International Journal of Geographical Information Science*, vol. 32, no. 3, pp. 601–621, 2018.
- [23] Y. Zhang, C. Shao, H. Li, and X. Ma, "Microscopic characteristics of lane-change maneuvers based on NGSIM," *Journal of Transport Information and Safety*, vol. 6, no. 33, pp. 19–32, 2015.
- [24] D. Levin, "The approximation power of moving least-squares," *Mathematics of Computation*, vol. 67, no. 224, pp. 1517–1531, 1998.
- [25] I. D. Gebru, X. Alameda-Pineda, F. Forbes, and R. Horaud, "EM algorithms for weighted-data clustering with application to audio-visual scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2402–2415, 2016.
- [26] P. Newson and J. Krumm, "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, Seattle, WA, USA, November, 2009.
- [27] W. C. Gao, G. L. Li, and N. Ta, "Survey of map matching algorithms," *Journal of Software*, vol. 29, no. 2, p. 26, 2018.
- [28] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [29] G. F. Jenks and F. C. Caspall, "Error on choroplethic maps: definition, measurement, reduction," *Annals of the Association of American Geographers*, vol. 61, no. 2, pp. 217–244, 1971.