Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 156296, 12 pages
http://dx.doi.org/10.1155/2014/156296

Research Article

Lane Detection in Video-Based Intelligent Transportation Monitoring via Fast Extracting and Clustering of Vehicle Motion Trajectories

Jianqiang Ren,[1,2] Yangzhou Chen,[1] Le Xin,[1] and Jianjun Shi[1]

[1] *College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China*
[2] *Department of Computer Science and Technology, Langfang Teachers University, Langfang 065000, China*

Correspondence should be addressed to Yangzhou Chen; yzchen@bjut.edu.cn

Received 19 April 2014; Revised 17 June 2014; Accepted 18 June 2014; Published 10 July 2014

Academic Editor: Qingsong Xu

Lane detection is a crucial process in video-based transportation monitoring system. This paper proposes a novel method to detect the lane center via rapid extraction and high accuracy clustering of vehicle motion trajectories. First, we use the activity map to realize automatically the extraction of road region, the calibration of dynamic camera, and the setting of three virtual detecting lines. Secondly, the three virtual detecting lines and a local background model with traffic flow feedback are used to extract and group vehicle feature points in unit of vehicle. Then, the feature point groups are described accurately by edge weighted dynamic graph and modified by a motion-similarity Kalman filter during the sparse feature point tracking. After obtaining the vehicle trajectories, a rough k-means incremental clustering with Hausdorff distance is designed to realize the rapid online extraction of lane center with high accuracy. The use of rough set reduces effectively the accuracy decrease, which results from the trajectories that run irregularly. Experimental results prove that the proposed method can detect lane center position efficiently, the affected time of subsequent tasks can be reduced obviously, and the safety of traffic surveillance systems can be enhanced significantly.

1. Introduction

Video-based traffic surveillance systems are widely used in traffic monitoring and management [1, 2]; they can supply more plentiful and more useful information than other sensor systems, for example, the ground induction loops, bridge sensors, and so on. Detection of lane position in the video scenes is a crucial basis of the traffic parameters extraction in lanes and the semantic analysis of traffic incident, vehicular behaviour, and so on [3]. Traditionally, the simplest method is specifying lane position manually during system installation. But this method is seldom used today because of its bad flexibility and error in the systems running. It cannot especially be suitable for pan-tilt-zoom (PTZ) cameras that are widely used in video-based traffic monitoring system today for its flexibility [4]. To a PTZ camera, parameters are allowed to adjust online. When these parameters change, the lanes position in video images must be redetected and the subsequent operations (such as traffic parameters extraction in lanes and the analysis of traffic incident and vehicular behaviour) will be affected or interrupted. The extraction accuracy and speed will directly affect the interrupted time of the subsequent operations.

Many researchers have been trying to study some automated detection methods. The existing methods include road-marking based methods, activity-map based methods, and vehicle-trajectories-clustering based methods. Lai and Yung [5] proposed a typical road-marking based lane detection method. In the method, the orientation discrimination and length of the traffic lane markings and curb structures are calculated and used to extract the lane position based on the fact that a lane always is the centre line of two parallel edge lines. Kim [6] used the random sample consensus with particle filter to extract lane marking and detected the lanes by using a probabilistic-grouping algorithm. Daigavane and Bajaj [7] proposed a hybrid approach based on edge detection. They use ant colony optimization to link disjointed edges that should be joined and use the Hough transform to

extract lanes. Although these methods have higher flexibility than the manual methods, they may lead to a sharp drop in accuracy and even to failure when the markings are blurred by contaminations or cannot even see in the dark, so they cannot apply to the cameras whose parameters might change in these cases. The activity-map based methods [8–10] can avoid being influenced by the clarity of road markings, but the lanes detecting results are rough. So they are not very suitable for the high accuracy requirements of lane-changing rate detection, high-level semantic analysis of traffic scene, and so on.

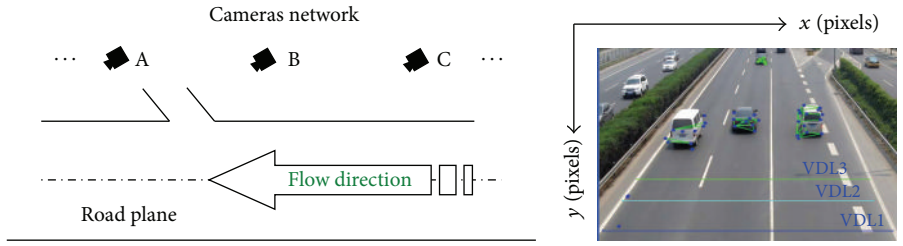
The vehicle-trajectories-clustering based methods can effectively overcome these weaknesses, but the existing methods almost have shortcomings in speed and accuracy. Several typical methods are summarized as follows. Hsieh et al. [11] used the background subtraction to extract vehicles and tracked them to obtain trajectories. Then the lanes are extracted via the histogram of different vehicles moving. Melo et al. [12] detected highway lanes based on global background subtraction and k-means cluster of trajectories. The trajectories are obtained by centroid tracking and represented by variable low degree polynomials. Then the lanes are extracted via a k-means cluster on the coefficient space. Xin et al. [13] used the foreground blobs tracking and multilayer spectral cluster to extract the vehicle trajectories and detect the traffic flow lanes. The speed and accuracy of these methods are not only related to the selected clustering method, but also related to the extraction method of vehicle trajectories. The shortcoming in speed results from the large computational cost of the selected global background subtracting and moving blobs tracking method, in which the whole image of video frames needs to be processed. The shortcoming in accuracy mainly comes from the influence of factors, for example, vehicles occlusion, moving shadows, and illumination sudden change, during the foreground blobs extraction.

Currently, except for the background subtracting and foreground blob tracking based methods which are selected in the above literatures, the feature-based tracking methods are increasingly used to extract vehicle trajectory, such as Coifman et al. [14] who developed a feature-based tracking system for detecting vehicles. In the process of vehicle trajectories extraction of this method, the segmentation among occluded vehicles is easier than above foreground blob tracking based methods, but only common motion constraint is used to group the features, so the accuracy of feature grouping is not very ideal when the occlusive vehicles have similar motion features. In addition, the feature tracking is not always robust over a long period of time, especially when vehicles turn or light changes suddenly. Kim [15] proposed an object detection and tracking method that combines dynamic feature grouping and background subtraction. The accuracy of feature grouping is improved, but the computation increases greatly. Kanhere and Birchfield [16] used the feature tracking with global background modeling to incrementally segment and track vehicles at low camera angles. Any feature that lies in background region is filtered out by using the foreground mask, and any feature that lies within a distance threshold from a background pixel is removed as shadow point. Then a plumb line projection is designed to group the features to yield the number and locations of the vehicles. This method can effectively eliminate the 3D perspective effects to vehicle location, but the use of global background subtraction significantly increases the computational burden. And the filtering error of shadow points is high and many good feature points that are near the edge of objects and should belong to foreground objects are mistakenly removed as shadow points, so the tracking effect is affected.

By analyzing the existing technology of lane detection and vehicle trajectories extraction, this paper proposes a fast-speed and high accuracy method of lane center detection based on rapid extracting and online clustering of vehicle motion trajectories, which can satisfy PTZ cameras as well as fixed cameras. Compared with the prior art, the proposed method has the advantages as follows. (1) It can be used both at night and in the daytime, and it has no limits of the clarity of road markings. (2) It is especially applicable for the challenging traffic scene of traffic congestions, in which the vehicle trajectories can be extracted well. (3) In respect of vehicle detection, we use three virtual detecting lines (VDLs) and VDL regions local background modeling to extract rapidly vehicle feature points and group them in unit of vehicle. The processing speed is greatly improved because the process avoids global background subtraction and feature point detection in the whole image. (4) The setting of the VDLs is automatically realized at upstream of traffic flow. We use the activity map to realize automatically the extraction of road region, the calibration of dynamic camera, and the setting of three VDLs which traverse the road. (5) A rough k-means incremental cluster of vehicle trajectories with Hausdorff distance is designed to realize the fast-speed and high accuracy online extraction of lane center, where the use of rough set reduces effectively the accuracy decreases resulting from the trajectories that run irregularly, which have similar distance to two neighbouring clusters and can be automatically distinguished from the regular trajectories by the clustering program. These irregular trajectories mainly include the trajectories with lane changes and the trajectories moving on the lane lines. This paper is organized as follows. Section 2 describes the system framework. Section 3 elaborates the incremental extracting and grouping of vehicle feature points. Section 4 presents the tracking and group correction of the feature points. Section 5 elaborates the extracting and clustering of vehicle trajectories. Section 6 illustrates the experimental results and analysis. Section 7 concludes this paper finally.

2. System Framework

Figure 1(a) shows the common cameras network in traffic monitoring systems, in which the camera (such as A, B, or C in the figure) is installed above the ground. Vehicles pass the scene from the close to the distant, and we can detect them when near before tracking them. Without loss of generality, we select the scene of camera B to introduce the proposed method because the scene not only includes straight lanes but also includes curving lane, so it is representative. The scene of camera B is shown in Figure 1(b).



(a) Location of cameras network (b) Video image of the scene of camera B

Figure 1: Sketch map of the system framework.

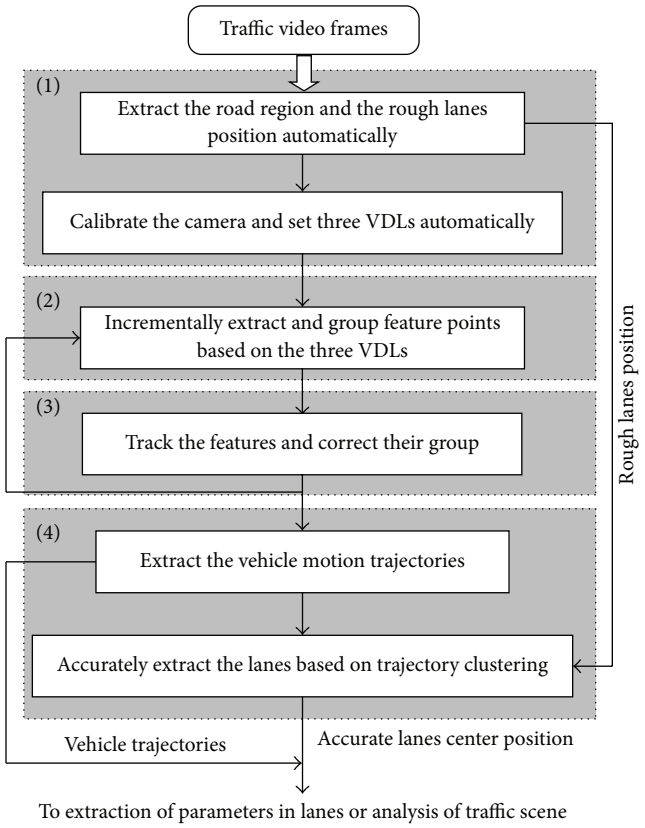


Figure 2: Block diagram of the proposed method.

When the camera parameters are changed online, the program based on our method is triggered by system signal. The process flow of our method mainly includes four steps, as is shown in Figure 2. Firstly, an activity map is created to detect the vehicle flow direction and realize automatically the calibration of dynamic camera, the extraction of road region, and rough lanes position; the methods can see [9, 10] and [17]. According to the results of dynamic camera calibration and extraction of road region, three VDLs are set automatically at the upstream of traffic flow, which is shown in Figure 1(b). The VDLs are set automatically to be perpendicular to the detected rough lanes and across the road region and the distance between them is equal according to the projection relationship of camera (such as 1/25 of the real length of the road region in experiments). The first VDL (VDL1) is as much as possible close to the edge of image. In addition, the result of rough lanes position is used as the initial cluster centres of the rough k-means cluster. Secondly, the three VDLs and a local background model with traffic flow feedback are used to extract and group vehicle feature points in unit of vehicle. Thirdly, the feature point groups are described accurately by edge weighted dynamic graph and modified by a motion-similarity Kalman filter during the sparse feature point tracking. At last, the vehicle trajectories are obtained, and a rough k-means incremental clustering with Hausdorff distance is designed to realize the rapid online extraction of lane center with high accuracy. We will discuss the second step to the last step in detail in the next couple of sections.

3. Incremental Extracting and Grouping of Vehicle Feature Points

This step is mainly based on the three VDL regions, which are set around the three VDLs, and their heights along y-axis are set to 7 pixels. In fact, experiments show that 7-pixel height can ensure the effect of background modeling and shadow eliminating. The quantity and quality of feature points also can be fully guaranteed by adding new points stepwise and rejecting unqualified points. In addition, some vehicles occlusion can be eliminated and the feature points can be grouped in unit of vehicle by using three foreground temporal-spatial images (FTSIs) [18], which are shown in Figure 3 and generated by accumulating VDL_i-locational foreground pixels in the order of time.

3.1. Modeling and Update of the Three VDL Regions Local Background. In order to reduce the influence of traffic congestions and moving shadows on accuracy of vehicular feature point detection, we design a dynamic background Gaussian mixture model with adaptive learning rate. For the traditional Gaussian mixture model, the learning rate α for update is always a fixed value ($0 \leq \alpha \leq 1$) [19–21]. Experiments show that $\alpha \in [0.001, 0.005]$ applies to the smooth traffic scenes. But during the rush hours, vehicle queue may happen and the vehicles may have slow speed or stop for a while in the VDL regions. If the learning rate is still fixed, these vehicles will be wrongly updated

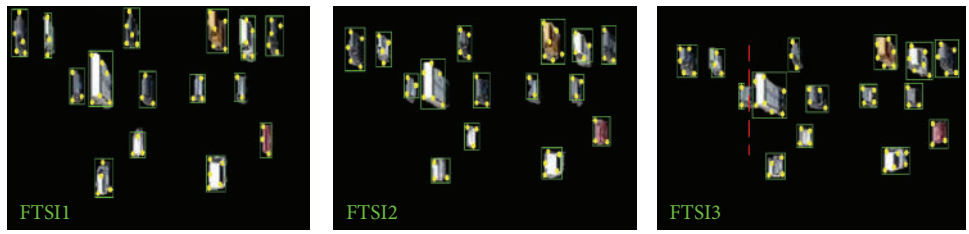


Figure 3: Results of extracting and grouping of feature points based on three FTSIs.

to the background. So we design a log-sigmoid dynamic adaptive update rate as follows based on experiments:

$$\alpha = \frac{\alpha_0}{1 + e^{-\beta(l-\xi)}}, \tag{1}$$

where α_0 is the update rate in smooth traffic scenes, β is the gradient parameter, l is the pixel-level distance between the vehicle queue tail and the Line3, and ξ is the pixel-level gap distance between neighbouring vehicles in queue, which is normally no longer than one vehicle. We use the average length of vehicles in the scene to estimate it, and the calculation based on FTSI_1 and FTSI_2 is as follows:

$$\xi = \frac{1}{m}\sum_{i=1}^{m}\frac{\Delta d \cdot w_i}{F_s \cdot \left(t_i^2 - t_i^1\right)}, \tag{2}$$

where m is vehicle blobs number in the FTSI, w_i is the width of the ith blob, and t_i^2 and t_i^1 are the t-coordinate of the ith blob centre in FTSI_2 and FTSI_1, respectively. Δd is the pixel-level distance between FTSI_1 and FTSI_2, and F_s is the video frame rate.

After that, the moving foreground in VDL regions can be extracted by using the local background subtraction. But there always are moving shadows in the extracted foreground, so we use the method in [22] to eliminate them.

3.2. Extracting and Grouping of Feature Points Based on Three FTSIs. The feature points are extracted initially from the foreground of VDL1 region and supplemented from the foreground of VDL2 region by using the good feature point automatic selection method in [23]. For the extracted feature points, though their relationship to vehicles has been recorded during the generation, some points of neighbouring vehicles may be wrongly grouped to the same vehicle since vehicles occlusion in the VDL region. This will affect the extracting accuracy of vehicle trajectories, so we improve the method in [24] to detect the occlusion and group the feature points in unit of vehicle based on the three FTSIs.

In the process, in order to improve the matching accuracy of vehicle blobs in different FTSI, the extracted feature points are tracked immediately by using the pyramidal KLT feature tracker [16] and the mean speed of the points that are considered to belong to the same vehicle is calculated as the approximation of the vehicle speed that is integrated into method [24] to match the vehicle blobs in different FTSIs. A merged blob in a FTSI will be separated if it disjoints in any other FTSIs. The results are as shown in Figure 3, in which the horizontal axis of FTSIs is time axis (t-axis) and the vertical axis is space axis (x-axis) corresponding to the horizontal axis of frame image. The two vehicles on the sides of the red line in FISI3 are separated by using this separating method.

4. Tracking and Group Correction of the Feature Points

Due to the impact of camera installation height, some vehicles may be occlusion in all the FTSIs and their feature points cannot be grouped into the correct vehicles only based on the three FTSIs, which is especially true in the heavy traffic; yet in these situations, the camera parameters are more likely to be changed to monitor the traffic scene in detail. So the wrong groups of feature points must be modified.

In our method, we firstly describe the point groups in unit of vehicle as the edge weighted dynamic graph $G = \{V, E, W\}$, where the V is the set of feature points in vehicle, E is the set of optimal edges between the feature points that are constructed by Delaunay triangulation method [25], and W is the set of the edge weight that is selected as the motion similarity of feature points.

Then the feature points are tracked to get trajectories; at the same time their groups are modified and filtered effectively based on the point motion similarity. The feature points tracking is based on the pyramidal Kanade-Lucas-Tomasi (KLT) tracker with local binary pattern (LBP) texture histogram. If the pyramidal KLT tracker is used alone, the feature points tend to drift when the vehicle speed is faster to the frame rate or the local luminance of the tracked vehicle changes suddenly in images. LBP feature has better robustness on these factors and has faster speed compared with traditional features [26], so we put the LBP texture histogram into the pyramidal KLT tracker to conquer these limitations. If a point position is changed in the new frame, the LBP texture histograms of them are calculated via a $7*7$ texture window and matched based on the Bhattacharyya distance; the tracking accuracy is enhanced obviously.

Meanwhile, considering that vehicles can be seen as rigid bodies, feature points in the same vehicle in image have high similar motion features, but the feature points in different vehicles do not. So we design a motion-similarity Kalman filter to modify and filter the point groups effectively during the points tracking, only in the direction of x-axis as an example since the analysis in the other direction of y-axis can be done in the same way. For two feature points i and j

of an edge in a frame image, let $X_{i,j}^{(t)}$ be their state vector $(\chi_{i,j}^{(t)}, v_{i,j}^{(t)}, a_{i,j}^{(t)})^T$, where $\chi_{i,j}^{(t)}$, $v_{i,j}^{(t)}$, and $a_{i,j}^{(t)}$ are their pixel-level distance alone x-axis, distance change rate, and distance change acceleration at the moment t, respectively. Then the model is as follows:

$$\begin{aligned} \widehat{X}_{i,j}^{(t)} &= \mathrm{A}X_{i,j}^{(t-1)} + \varsigma, \\ Z_{i,j}^{(t)} &= \mathrm{H}\widehat{X}_{i,j}^{(t)} + \eta, \end{aligned} \tag{3}$$

where $\widehat{X}_{i,j}^{(t)}$ is the prediction vector, $Z_{i,j}^{(t)}$ is the measurement vector, ς is the system noise vector, and η is the measurement noise vector; the transition matrix A and the observation matrix H are as follows:

$$\mathrm{A} = \begin{bmatrix} 1 & \tau & \dfrac{\tau^2}{2} \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathrm{H} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \tag{4}$$

where τ is the sampling period.

At the moment t, the difference vector of the measurement value of $X_{i,j}^{(t)}$ and the predicted values of it are represented as $\Delta_{i,j}^{(t)}$. If the first component of the $\Delta_{i,j}^{(t)}$ is not greater than zero, the estimation of system state vector is updated as follows:

$$X_{i,j}^{(t)} = \widehat{X}_{i,j}^{(t)} + K_k \Delta_{i,j}^{(t)}, \tag{5}$$

where K_k is the Kalman gain [27].

Otherwise, the motion similarity of points i and j will be judged to be too poor and their $w_{i,j}$ in the edge weighted dynamic graph G will be set to 0; the point groups can be modified. If the point number of a modified group is less than 3, it will be judged as the result of tracking drift and be removed.

5. Extracting and Clustering of Vehicle Trajectories

The modified feature point groups can accurately express the vehicles, and the point trajectories have been obtained during tracking. But if we cluster the point trajectories directly, the calculation is too complex and the speed is very slow. In our method, the vehicle trajectories are extracted from the point trajectories and then the lanes are detected based on the vehicle-trajectories-clustering.

5.1. Extracting of Vehicle Trajectories. For the edge weighted dynamic graph $G = \{V, E, W\}$ of each vehicle, the modified results of the component V are the set of stable feature points; E is the set of optimized edges of each feature point in V. Considering the fact that the feature point in G is often closer to the vehicular center if it has more edges in E, this paper uses the weighted average method to calculate the distribution center $P(\overline{x}, \overline{y})$ of the vehicle points as follows:

$$\overline{x} = \frac{1}{2n} \sum_{1 \le i \le n} m_i x_i, \qquad \overline{y} = \frac{1}{2n} \sum_{1 \le i \le n} m_i y_i, \tag{6}$$

where n is the size of V, x_i and y_i are the image coordinates of the ith feature point, and weight m_i is the edge number of the ith point.

Then, we calculate the Euclidean distance $d_{i,p}$ between each feature point i in V and $P(\overline{x}, \overline{y})$, rearrange the feature points according to their $d_{i,p}$ from small to large, and extract the first 3 points to constitute the center nearest neighbor point set C. Then the jth point trajectory is extracted as the vehicle trajectory as follows:

$$j = \underset{c \in C}{\operatorname{argmax}}\ l_c, \tag{7}$$

where l_c is the trajectory length of the point c. In other words, the longest trajectory of the points in C is extracted as the vehicle trajectory.

5.2. Extraction of Lane Center Based on Clustering of Vehicle Trajectories. Based on the vehicle trajectories, the lane center can be extracted via online incremental clustering. Considering that there always are trajectories that run irregularly in video sequence (which mainly include the trajectories with lane changes and the trajectories moving on the lane lines), if they are abruptly relegated to one cluster, the clustering accuracy would be influenced. Rough set theory as a forceful tool in processing uncertainties can resolve the problem well. So we design a rough k-means clustering method with Hausdorff distance to realize the rapid online extraction of lane center with high accuracy. When the camera parameters are changed online, the vehicle trajectories will be extracted, and the incremental clustering program (see Section 5.2.2) will be triggered at regular intervals to run in the mode of initial clustering and incremental correction clustering. The first running of the incremental clustering program is called the "initial clustering," which is based on the initial cluster centres (see Section 5.2.1) and outputs the "initial clustering results" by clustering the vehicle trajectories in the first time interval. The second running of the program is called the "first incremental correction," which corrects the initial clustering results and outputs the "first incremental correction results" by clustering the vehicle trajectories in the second time interval. The third running of the program is called the "second incremental correction," which corrects the first incremental correction results and outputs the "second incremental correction results" by clustering the vehicle trajectories in the third time interval. Analogously, the incremental clustering program will continuously run many times until the adjacent two results have no longer obvious change. Then the stable results are extracted as the final positions of the lanes center.

5.2.1. Selection of k, Initial Cluster Centres, and Characteristics. For a traffic scene, the number of lanes (i.e., k) is constant, so it can be set manually. The initial cluster centres are selected as the result of rough lanes position, which are extracted by using the activity map in step 1. The random selection of initial centres is avoided and the clustering performance is improved.

For characteristics, in order to improve the method robustness, we select Hausdorff distance between the trajectory and the centres. For a trajectory and the current centre, let A_i be the shorter one and A_j the longer one; then their Hausdorff distance is as follows:

$$h\left(A_i, A_j\right) = \frac{1}{N_i} \sum_{a \in A_i} \min_{b \in A_j} \left\| \left(x^a - x^b, y^a - y^b\right) \right\|, \quad (8)$$

where N_i is the length in pixels of A_i and (x^a, y^a) and (x^b, y^b) are the coordinates of point a and point b in A_i and A_j, respectively.

5.2.2. Rough k-Means Incremental Clustering. In the rough k-means clustering, each lane cluster is expressed by an upper approximation and a low approximation. The latter is a subset of the former and their set difference is called boundary area. The low approximation is used to save the vehicle trajectories belonging certainly to a lane cluster, that is, the trajectories of the conventional vehicle in the lane. The boundary area is used to save the trajectories that run irregularly. In calculation of the new center, trajectories in the lower approximation should be given larger weights than those in the boundary area.

Firstly, we set the cluster number k and the initial centers $c_1, c_2, \ldots, c_k$, then calculate the Hausdorff distance $h(A_x, c_i)$ $(i = 1, 2, \ldots, k)$ between every trajectory A_x and every center c_i, and find the cluster m with the shortest distance as follows:

$$h\left(A_x, c_m\right) = \min_{i=1,2,\ldots,k} \left(h\left(A_x, c_i\right)\right). \quad (9)$$

Then

$$\begin{aligned} A_x \in \overline{C_m} \wedge A_x \in \overline{C_j}, \quad j \neq m \wedge \frac{h\left(A_x, c_j\right)}{h\left(A_x, c_m\right)} \leq \lambda \\ A_x \in \underline{C_m}, \quad \text{otherwise,} \end{aligned} \quad (10)$$

where $1 \leq \lambda \leq 1.5$, $\overline{C_m}$ and $\underline{C_m}$ are the upper and lower approximation of the mth cluster, and $\underline{C_m} \subseteq \overline{C_m}$.

After that, every cluster centre is updated as follows:

$$c_i = \begin{cases} \dfrac{w_l \sum_{A_x \in \underline{C_i}} A_x}{\left|\underline{C_i}\right|} + \dfrac{(1 - w_l) \sum_{A_x \in (\overline{C_i} - \underline{C_i})} A_x}{\left|\overline{C_i} - \underline{C_i}\right|}, & \overline{C_i} \neq \underline{C_i}, \\ \dfrac{\sum_{A_x \in \underline{C_i}} A_x}{\left|\underline{C_i}\right|}, & \text{otherwise,} \end{cases} \quad (11)$$

where $i = 1, 2, \ldots, k$, the weight $w_l \in [0.5, 1]$, and $|\cdot|$ is the cardinality.

The clustering is an iterative process from (9) to (11), which does not stop until the result centres between two adjacent iterations have no obvious difference.

6. Experiment Results and Analysis

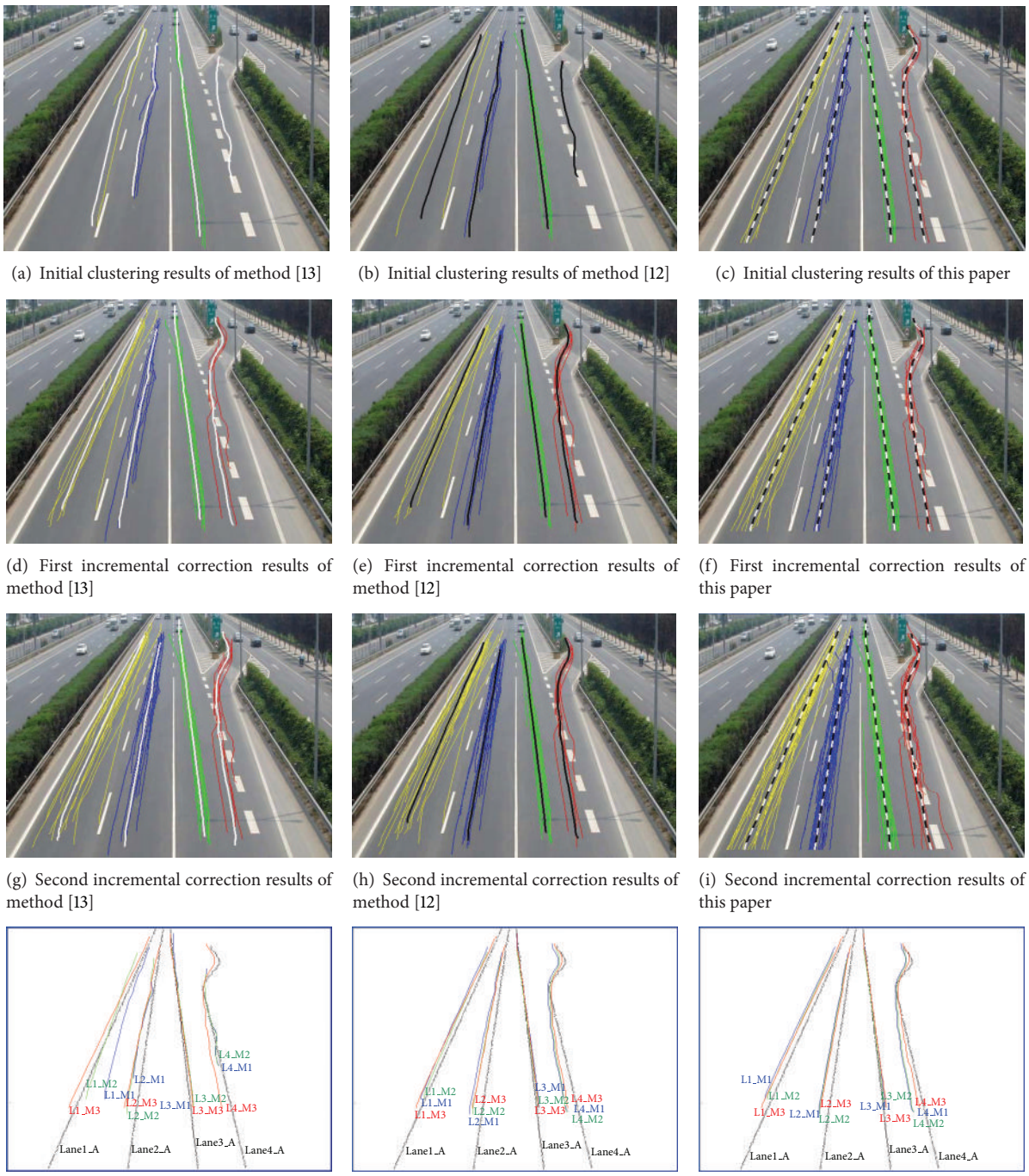
6.1. Test Data and Parameter Settings. A large number of experiments have been done based on a 270 minutes traffic video material set, which includes nine 30-minute videos collected at an exit road segment near Houfeng Bridge in Beijing. These videos are collected by a PTZ camera during morning rush hours, evening rush hours, and normal traffic hours for three consecutive days. And the camera parameters are changed every 10 minutes manually. These test videos have 640×480 resolution at a frame rate of 25 frames per second.

Our method is implemented with C++ programming language by using multithreading technology and runs on a computer with 3.30 GHz dual-core CPU and 4 GB RAM. In the main thread, vehicle trajectories are detected based on vehicular feature point detecting and grouping, and the results are exported to a buffer queue. The child thread will be triggered when the camera parameters are changed online. Then it obtains the new trajectories from the buffer queue and achieves the incremental clustering every 40 seconds based on the last results until the results have no longer obvious change or the camera parameters are changed again.

6.2. Comparison and Analysis of Experimental Results. In order to analyze the performance of our method, we compare and analyze the detecting results of lanes with those obtained in existing method [12] and method [13].

6.2.1. Qualitative Analysis. In every process of lane detection after the camera parameters changed, the initial clustering is executed based on the initial cluster centres firstly. Then the incremental clustering is executed every 40 seconds based on the last result. Since the initial clustering and the subsequent two times of the incremental clustering best demonstrate the method speed and accuracy, we use them to compare the three methods as shown in Figure 4, in which the video is collected from 11:50 on June 14, 2013, and the traffic scene in this period has delegation denotation in respect of traffic flow, vehicular speed, and so on. The four lanes in the scene are named as lane1 to lane4 from left to right.

Figures 4(a)–4(c) show the initial clustering results of the three methods. For method [13], although four lanes are detected, the error is large. Lane1 result significantly deviates from the actual centre and is close to lane2. Lane2 and lane4 are close to the actual centres but their length is too short. For method [12], the results of lane1 to lane3 are better than those of method [13], but the result of lane4 is too short too, which is mainly influenced by the speed of background modelling and trajectories extracting. For our method, though the accuracy is not good enough, especially the tail end of lane4, the results are better than the other methods obviously. Figures 4(d)–4(f) show the results of the first incremental correction. We can see that the results are all better than those of Figures 4(a)–4(c). For method [13], the relatively complete results of the four lanes are extracted but the error is still larger. For method [12], the results are better than method [13]. For our method, the tail end of lane4 is corrected and each lane extraction results are closer to



(a) Initial clustering results of method [13]

(b) Initial clustering results of method [12]

(c) Initial clustering results of this paper

(d) First incremental correction results of method [13]

(e) First incremental correction results of method [12]

(f) First incremental correction results of this paper

(g) Second incremental correction results of method [13]

(h) Second incremental correction results of method [12]

(i) Second incremental correction results of this paper

(j) Results comparison of the initial clustering

(k) Results comparison of first incremental correction

(l) Results comparison of second incremental correction

FIGURE 4: Results comparison after the first three clusters.

the actual position. Figures 4(g)–4(i) show the results of the second incremental correction. All of the lanes are extracted more accurately by all of the three methods. The results of method [12] are better than those of method [13]. All the four lanes results of our method are extracted with the most accuracy and the tail end of lane4 is very close to the actual position.

6.2.2. Quantitative Analysis. We extracted manually the actual position of lanes based on the pavement markings. Each extracted actual lane runs through the visible surface of the corresponding lane in length. Then, we compared the three methods results with the actual lanes and analyzed their accuracy quantitatively. For a good method, the lane extracting results not only should approach the actual station

Table 1: AEP results of the three methods to actual lanes.

Count of clustering	Methods	AEP (pixels)				
		Lane 1	Lane 2	Lane 3	Lane 4	Average
Initial clustering	literature [13]	18015.62	18293.95	8777.63	18779.82	15841.76
	literature [12]	12369.46	11593.03	8400.65	19184.01	12486.79
	This Paper	8956.31	9917.08	8015.73	9230.62	9029.94
1st incremental correction	literature [13]	11696.72	10149.67	8307.06	9675.95	9900.86
	literature [12]	11235.56	9396.41	8081.10	9283.25	9555.57
	This Paper	8904.35	9436.90	7927.08	8861.32	8775.64
2nd incremental correction	literature [13]	12142.81	9411.09	7899.97	9684.26	9791.31
	literature [12]	11783.82	9529.97	7880.56	9201.17	9673.26
	This Paper	8755.62	8965.90	7549.95	7451.58	8180.76

as possible, but also should be as complete as possible in length, so we use (12) to analyze the absolute error in pixel-level (AEP) of the ith lane between the method result and the actual position. Consider

$$\mathrm{AEP}\left(L_i, \widehat{L}_i\right) = \sum_{a \in \widehat{L}_i} \min_{b \in L_i} \left\| \left(x_a - x_b, y_a - y_b\right) \right\|, \tag{12}$$

where $\widehat{L}_i$ is the actual position and L_i is the method result.

Figures 4(j)–4(l) compare the results of the three methods with the actual lanes during the initial clustering and the two incremental corrections. Where lane1_A to lane4_A are the actual position of the four lanes, L1_M1 to L4_M1, L1_M2 to L4_M2, and L1_M3 to L4_M3 are the detection results of the four lanes via method [13], method [12], and our method. According to (12), the values of AEP are as shown in Table 1.

Analysis shows that, for the initial clustering results, all lane results in our method are less than those of others; the error of every lane in method [13] is larger than that in method [12] except for lane4. The average of AEP of all lanes in our method is less 6811.82 pixels than method [13] and less 3456.85 pixels than method [12]. For the first incremental correction, the differences of the average results between the three methods are reduced. Our method is better than others except for in lane2. After the second incremental correction, the average of AEP of our method is only 8180.76 pixels, which is less 16.5% and 15.4% than the others. This mainly benefits from the fast speed of vehicle trajectory extracting. In the main thread, to average computing time per frame, the incremental extracting and grouping of vehicle feature points take about 29 ms, and the tracking and group correction of the feature points take about 131 ms. The average speed is about 25% faster than that of the methods based on traditional global background subtraction and is about 43% faster than that of the methods based on traditional feature point detection in the whole image.

6.3. Statistical Analysis of Experimental Results. We take the clustering process after camera parameters change as the unit to analyze statistically the mean of AEP (MAEP) of all the experimental data via the three methods, which is calculated as follows:

$$\mathrm{MAEP}\left(L_i, \widehat{L}_i\right) = \frac{1}{n} \sum_{n} \left(\frac{1}{\widehat{N}_i} \mathrm{AEP} \right), \tag{13}$$

where n is the total number of experiments and $\widehat{N}_i$ is the pixel-level length of $\widehat{L}_i$.

MAEP of the initial clustering and all of the incremental corrections are shown in Figure 5, where the abscissa of each graph is set to the count of clustering, in which the initial clustering is represented as "1," the first incremental correction is represented as "2," and so on.

From Figure 5, we can see that all of the three methods have better convergence but the convergence speed of our method is the fastest and the convergence process is smoother than other methods. For the results of lane1 in Figure 5(a), MAEP of the initial clustering in method [13] is the largest. Though it has better convergence, it oscillates weakly in the whole process. Method [12] is basically stable at near 22 after 9 times clustering. For our method, MAEP of the first clustering and the average of MAEP of the whole process are all less than those of others. Only after 5 times of clustering, our method is basically stable at near 21 pixels. For lane2 in Figure 5(b), MAEP of our method is also the minimum in the first seven clusters, method [12] followed, and method [13] is the maximum. And the smoothness of method [13] is better than that of lane1. After 7 times of clustering, MAEP of our method is basically stable at about 20. For lane3 in Figure 5(c), all MAPE are relatively small and that of our method is the minimum in the first 6 times of clustering results. The result of method [12] is slightly smaller than our method after the seventh clustering. For lane4, the performance of our method is the best and the MAEP is basically stable at about 20 only after four clusters.

To sum up, our method has better performance in respect of speed and average accuracy. In addition, the MAEP result of each lane via all the methods is stable at a nonzero value, and the MAEP of lane1 and lane4 are greater than that of lane2 and lane3. These are mainly due to the projection effects of vehicle height.

6.4. Statistical Analysis of the Results of Extensive Testing in Different Scenes. In order to test our method performance in

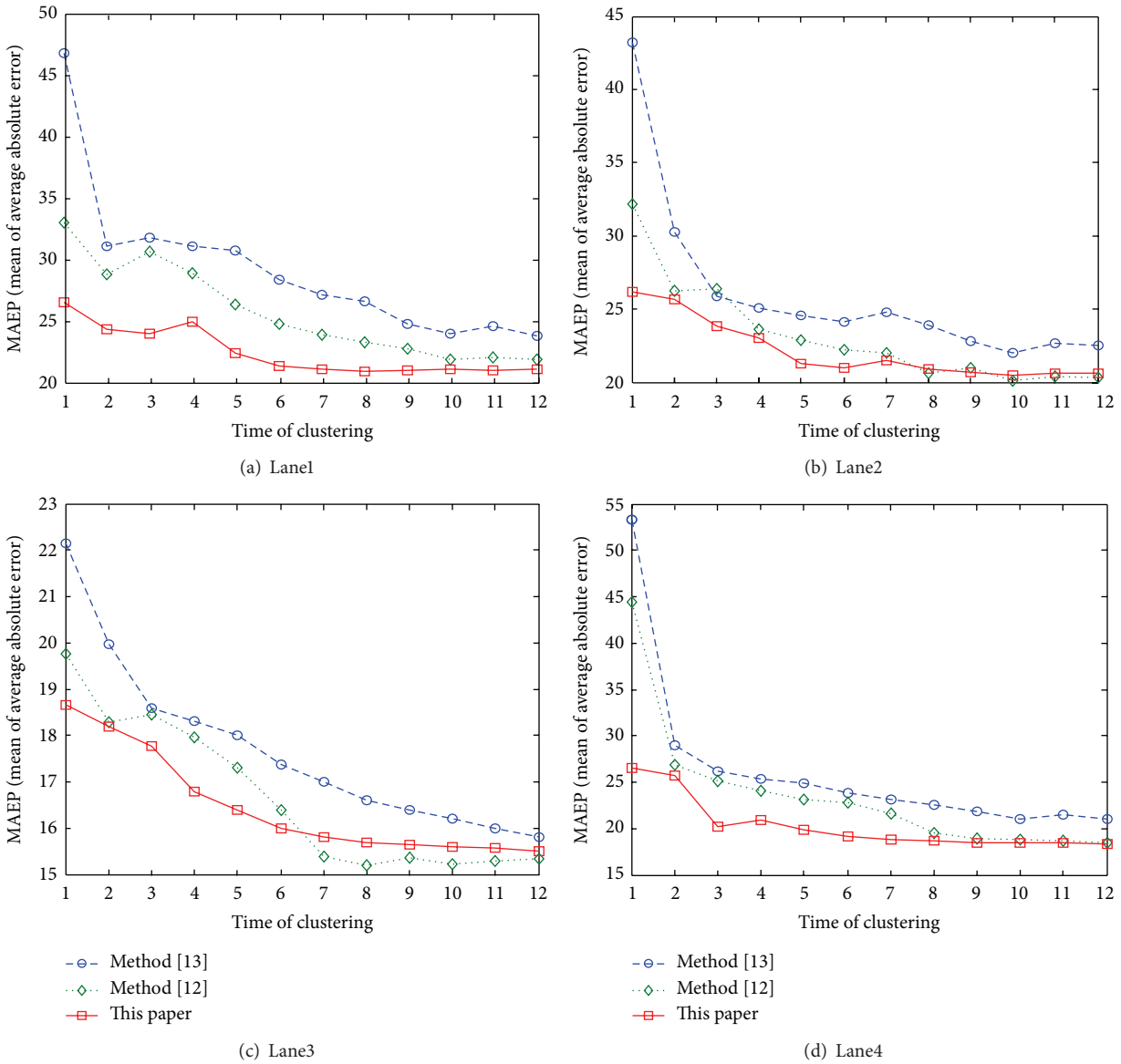


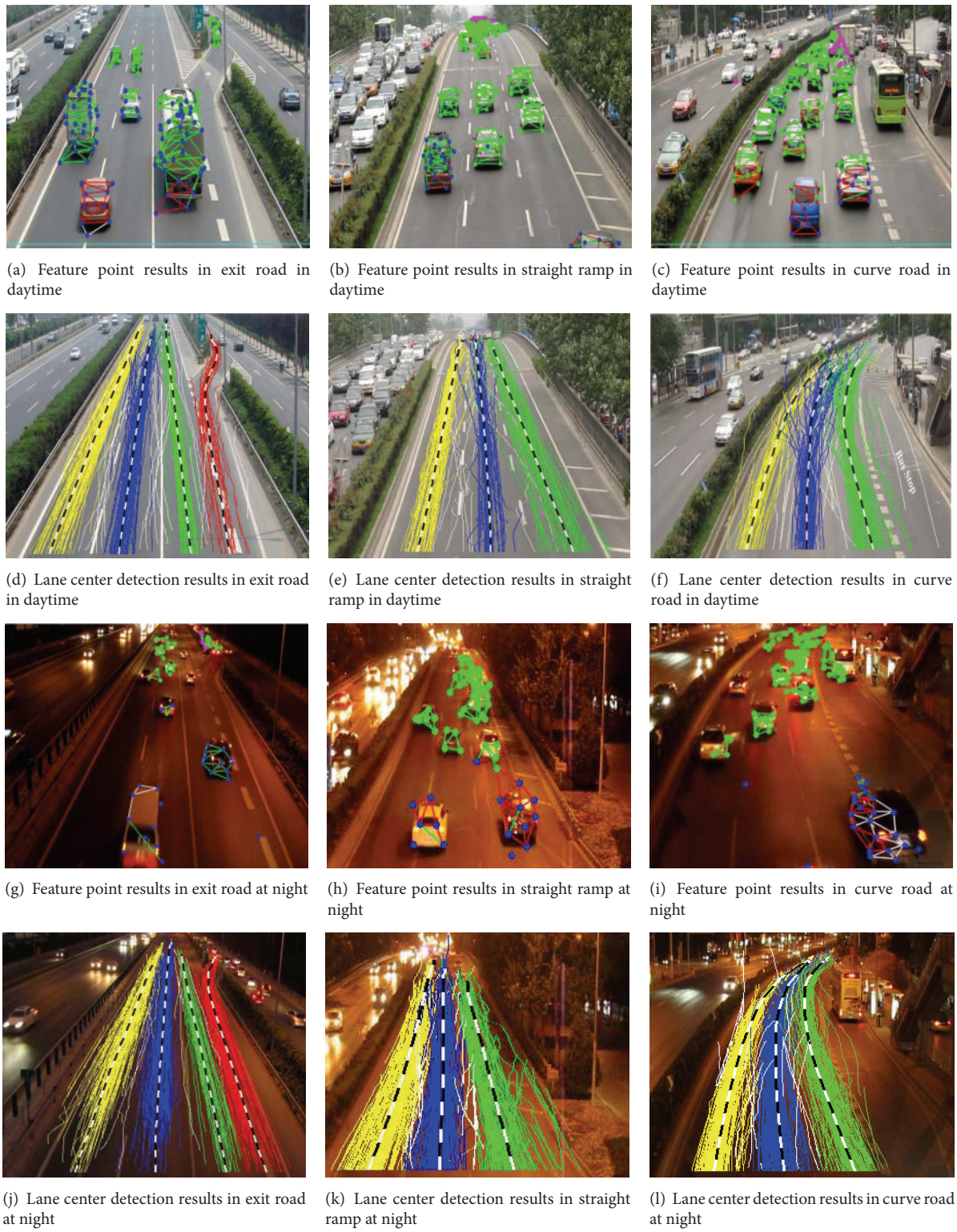
Figure 5: MAEP curves of the initial clustering and all the incremental corrections.

different scenes, we also collected 120 minutes traffic videos from two other typical road segments and tested our method based on all of them. Figure 6 shows the typical sample results of detecting and grouping of feature points and the results of lanes detection based on the 270-minute videos above at the exit road and the 120-minute new videos at the straight ramp and the curve road. We can see that our method has high accuracy not only in the exit road but also in the straight ramp and the curve road. Though the road distortion in frame image that results from the influence of gradient is obvious in the straight ramp, and there are lots of vehicles and their trajectories that are messy in the curve road, all the lanes can be extracted well. Our method has high robustness to all the scenes.

All the experimental data of the three methods based on all the videos in the three scenes are analyzed statistically. The overall average MAPE of method [13], method [12], and this paper are 34.72, 29.23, and 23.96 pixels, respectively. In general, our method is the most accurate and robust in all the methods, which mainly benefits from two factors. The first is extraction efficiency of the vehicle trajectory in the method, which can provide the trajectory samples as much as possible for each clustering operation. The second is the use of rough k-means clustering method, in which the trajectories that run irregularly are consigned to the boundary area of lanes in the clustering process, as is shown by white trajectory lines in Figures 6(d)–6(f) and 6(j)–6(l), and their influence on the accuracy of lane detection is reduced obviously.

7. Conclusion

This paper proposed a novel method of lane center detection via vehicle motion trajectories. Firstly, the rapid extraction



(a) Feature point results in exit road in daytime

(b) Feature point results in straight ramp in daytime

(c) Feature point results in curve road in daytime

(d) Lane center detection results in exit road in daytime

(e) Lane center detection results in straight ramp in daytime

(f) Lane center detection results in curve road in daytime

(g) Feature point results in exit road at night

(h) Feature point results in straight ramp at night

(i) Feature point results in curve road at night

(j) Lane center detection results in exit road at night

(k) Lane center detection results in straight ramp at night

(l) Lane center detection results in curve road at night

Figure 6: Testing results in the typical road segments.

and grouping of vehicular feature points are realized based on three VDLs and three FTSIs, the global background subtracting is avoided, and the detection speed is enhanced. Then, the feature points are tracked by the pyramidal KLT tracker with LBP texture histogram, and their groups are described by edge weighted dynamic graph and modified by a motion-similarity Kalman filter. Finally, the vehicle trajectories are detected from the feature point trajectories and the lanes are extracted rapidly and high accurately by using a rough k-means incremental cluster with Hausdorff distance.

The experiments proved the feasibility and efficiency of our method. The lane results can support the extraction of traffic parameters in lanes and analysis of the traffic incident, vehicular behaviour, and so on. The affected time of these tasks after camera parameters changed online can be reduced obviously and the safety of traffic surveillance systems can be enhanced significantly.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61273006), National High Technology Research and Development Program of China (863 Program) (Grant no. 2011AA110301), and Science and Technology Support Foundation of Hebei Province (Grant no. 13210807) and Scientific Research Foundation of Hebei Higher Education (Grant No. Z2011118).

References

[1] Y. Lipetski, G. Loibner, M. Ulm, W. Ponweiser, and O. Sidla, "Real-time traffic jam detection and localization running on a smart camera," in *Video Surveillance and Transportation Imaging Applications*, vol. 9026 of *Proceedings of the SPIE*, 2014.

[2] L. Li, *Video based road traffic monitoring [M.S. dissretation]*, Faculty of the Engineering and Physical Science, University of Manchester, Manchester, NH, USA, 2013.

[3] T. J. Gates, P. T. Savolainen, T. K. Datta et al., "Use of both centerline and shoulder rumble strips on high-speed two-lane rural roadways," in *Transportation Research Record: Journal of the Transportation Research Board, No. 2301*, vol. No. 2301, pp. 36–45, Transportation Research Board of the National Academies, Washington, DC, USA, 2012.

[4] K. T. Song and J. C. Tai, "Dynamic calibration of pan-tilt-zoom cameras for traffic monitoring," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 36, no. 5, pp. 1091–1103, 2006.

[5] A. H. Lai and N. H. Yung, "Lane detection by orientation and length discrimination," *IEEE Transactions on Systems, Man and Cybernetics B: Cybernetics*, vol. 30, no. 4, pp. 539–548, 2000.

[6] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 16–26, 2008.

[7] P. M. Daigavane and P. R. Bajaj, "Road lane detection with improved canny edges using ant colony optimization," in *Proceedings of the 3rd IEEE International Conference on Emerging Trends in Engineering and Technology*, pp. 76–80, 2010.

[8] B. D. Stewart, I. Reading, M. S. Thomson, T. D. Binnie, K. W. Dickinson, and C. L. Wan, "Adaptive lane finding in road traffic image analysis," in *Proceedings of the 7th International Conference on Road Traffic Monitoring and Control*, pp. 133–136, April 1994.

[9] T. N. Schoepflin and D. J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 2, pp. 90–98, 2003.

[10] T. N. Schoepflin, D. J. Dailey, and P. Briglia, *Algorithms for estimating mean vehicle speed using uncalibrated traffic management cameras, [Doctoral dissertation]*, Washington State Department of Transportation, Washington, Wash, USA, 2003.

[11] J. Hsieh, S. Yu, Y. Chen, and W. Hu, "Automatic traffic surveillance system for vehicle tracking and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 179–183, 2006.

[12] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, "Detection and classification of highway lanes using vehicle motion trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 188–200, 2006.

[13] L. Xin, D. L. Yang, Y. Z. Chen, and Z. Li, "Traffic flow characteristic analysis at intersections from multi-layer spectral clustering of motion patterns using raw vehicle trajectory," in *Proceedings of the 14th IEEE International Intelligent Transportation Systems Conference (ITSC '11)*, pp. 513–519, IEEE, Washington, DC, USA, October 2011.

[14] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research C: Emerging Technologies*, vol. 6, no. 4, pp. 271–288, 1998.

[15] Z. W. Kim, "Real time object tracking based on dynamic feature grouping with background subtraction," in *Proceeding of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[16] N. K. Kanhere and S. T. Birchfield, "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 148–160, 2008.

[17] T. N. Schoepflin and D. J. Dailey, "Correlation technique for estimating traffic speed from cameras," *Transportation Research Record*, vol. 1855, no. 1, pp. 66–73, 2003.

[18] J.-Q. Ren, L. Xin, Y.-Z. Chen, and D.-L. Yang, "High-efficient detection of traffic parameters by using two foreground temporal-spatial images," in *Proceedings of the 16th IEEE International Conference on Intelligent Transportation Systems*, pp. 1965–1970, 2013.

[19] H. H. Lin, J. H. Chuang, and T. L. Liu, "Regularized background adaptation: a novel learning rate control scheme for Gaussian mixture modeling," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 822–836, 2011.

[20] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 246–252, June 1999.

[21] Y. L. Tian, A. Senior, and M. Lu, "Robust and efficient foreground analysis in complex surveillance videos," *Machine Vision and Applications*, vol. 23, no. 5, pp. 967–983, 2012.

[22] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 918–923, 2003.

[23] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593–600, June 1994.

[24] N. C. Mithun, N. U. Rashid, and S. M. M. Rahman, "Detection and classification of vehicles from video using multiple time-spatial images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1215–1225, 2012.

[25] L. J. Guibas, D. E. Knuth, and M. Sharir, "Randomized incremental construction of Delaunay and Voronoi diagrams," *Algorithmica*, vol. 7, no. 4, pp. 381–413, 1992.

[26] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[27] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering D*, vol. 82, pp. 35–46, 1960.