



Vehicle Behavior Analysis Using Target Motion Trajectories

Huan-Sheng Song, Sheng-Nan Lu, Xiang Ma, Yuan Yang, Xue-Qin Liu, and Peng Zhang

Abstract—In this paper, a real-time vehicle behavior analysis system is presented, which can be used in traffic jams and under complex weather conditions. In recent years, many works based on background estimation and foreground extraction for traffic event detection have been reported. In these studies, the vehicle images need to be accurately segmented, although uneven illumination, shadows, and vehicle overlapping are difficult to handle. The main contribution of this paper is to make a point tracking system for vehicle behavior analysis without a difficult image segmentation procedure. In the proposed system, feature points are extracted using an improved Moravec algorithm. A specially designed template is used to track the feature points through the image sequences. Then, trajectories of feature points can be obtained, whereas unqualified track trajectories are removed using decision rules. Finally, the vehicle behavior analysis algorithms are applied on the track trajectories for traffic event detection. The proposed system has been used widely by Chinese highway management departments. The application performances show that the newly developed system and its algorithms are robust enough for vehicle behavior analysis under complex weather conditions.

Index Terms—Corner detection, tracking, traffic event detection, vehicle behavior analysis.

I. INTRODUCTION

OVER the past several years, video surveillance has been a key part of intelligent transportation systems (ITSs), and it provides a rich information source for human understanding. Extracting and analyzing visual information from a traffic surveillance video has gained a great amount of interest worldwide [1]. The goal of video-based automatic traffic analysis is to detect and track the vehicles that move through a controlled area and to identify abnormal events, such as traffic congestion, speeding violations, illegal driving behavior, and

accidents. Through video monitoring and analysis, it is possible to compute traffic parameters such as the speed, flow rate, and density of vehicles [2]. Hence, the vehicle behavior analysis is essential for daily traffic management. It provides an advanced and feasible surveillance scheme for making an immediate response when abnormal events occur.

Successful video-based systems for traffic monitoring must be adaptive to different weather and illumination conditions. Many problems such as illumination variation, shadow effects [19], and vehicle overlapping that appear in traffic jams create many difficulties [18] in analysis. For example, shadows that always accompany moving vehicles are likely to be misidentified as a part of the vehicle and will result in incorrect segmentation. At night, vehicle headlights and bad illumination can also cause many problems for detection. Therefore, the vehicle behavior analysis under such situations is always an important but challenging subject.

Several studies have been published concerning traffic event detection. Ikeda *et al.* [11] applied the background difference and time difference to extract vehicles and to estimate the distance between two vehicles. In this way, the speed, object size, and trajectory can be obtained from the position and time differences. All the measurement functions work successfully in passing judgment on incident occurrences, such as vehicles stopping, moving slowly, or changing lanes, as well as objects falling. Wu *et al.* [3] developed vehicle and incident detection in their system. Background extraction and lane detection were first used to get the background image and lane information, and then moving vehicles were tracked. Thus, vehicles that changed lanes or stopped would be detected by the vehicle tracking with lane information. This approach could be also used to evaluate traffic flow or detect a congestion condition. Ki and Lee [4] used a frame difference method to extract moving objects. Their model was adopted for decision-making based on the extracted features, e.g., the variation rate of velocity, position, area, and direction of moving objects. Kamijo and Inoue [10] used the spatiotemporal Markov random field model to segment an object, and the edge map is also used in the detection system. The system was capable of checking the behavior of lane changes, but the lane information was not detected automatically. Lin *et al.* [15] presented a traffic monitoring system using active-contour model tracking algorithms, and the stand-alone image tracker extracted traffic parameters and detected car accidents from image sequences. Hu *et al.* [21] reported a method of traffic accident prediction using 3-D model-based vehicle tracking. A fuzzy self-organizing neural network algorithm was then applied to learn activity patterns from sample trajectories. The vehicle activity was predicted by

Manuscript received December 22, 2012; revised April 20, 2013, July 20, 2013, October 16, 2013, and January 13, 2014; accepted February 14, 2014. Date of publication March 11, 2014; date of current version October 14, 2014. This work was supported in part by the National High Technology Research and Development Program of China under Grant 2012AA112312, by the Fundamental Research Funds for the Central Universities of China under Grant CHD2011TD012, and by the Research Fund for the Doctoral Program of Higher Education of China under Grant 20120205110001. The review of this paper was coordinated by Dr. M. S. Ahmed.

H.-S. Song, X. Ma, Y. Yang, and X.-Q. Liu are with the School of Information Engineering, Chang'an University, Xi'an 710064, China (e-mail: hshsong@chd.edu.cn; maxiangmail@163.com; 291226199@qq.com; 495676522@qq.com).

S.-N. Lu is with the School of Information Engineering, Chang'an University, Xi'an 710064, China and also with the School of Computer, Xi'an Shiyou University, Xi'an 710065, China (e-mail: lushengnan@xsyu.edu.cn).

P. Zhang is with the China Highway Engineering Consulting Group Company Ltd., Beijing 10088, China (e-mail: 82699755@qq.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVT.2014.2307958

locating and matching each trajectory with the learned activity patterns, and then, the occurrence probability of a certain traffic accident was determined. As a result, detailed geometric data of vehicles were required, and the computing complexity was high.

From the aforementioned systems, it is noticed that the traffic events were usually detected by tracking vehicle trajectories, which was based on accurate segmentation for moving vehicles [17], [22]. This approach was defined as a top-down approach by Buch *et al.* [12]. As usual, image segmentation is one of the problems in analyzing the computer vision, particularly under various complex traffic environments, such as cloudy, rainy, and snowy conditions. Many problems such as illumination variations, shadow effects [19], and vehicle overlapping that appear in traffic jams will make the task of segmenting individual vehicles difficult [18]. Meanwhile, the complex algorithms of high-level vehicle modeling and learning create a great amount of computation, which in turn is mostly unstable and time-consuming.

In this paper, different from the idea of tracking objects as a whole, we make an effort to select the most common low-level features on vehicles for tracking without segmentation. Such approaches have been proposed in some publications. Beymer *et al.* [13] presented a real-time computer vision system based on the Harris corner [14] tracking algorithm. These corners were clustered into vehicles based on proximity and similar 2-D motion, and the system was used for estimating traffic parameters such as flow, velocity, and density. Similarly, Kanhere and Birchfield [16] tracked points. In their research [16], the points were back-projected into 3-D before clustering into vehicles. Unfortunately, these methods have a rather limited capability to analyze traffic behaviors because traffic event detection needs highly accurate tracking trajectories, which may cost more time and computer resources.

Point features have the advantage of efficient locating and easy tracking. Hence, we suggested a new traffic event detection algorithm using the point features and developed a system for automatically detecting and reporting the traffic conditions. The significance of this system lies in its temporal processing for traffic parameter evaluation and traffic behavior identification, and accordingly, it works well for solving many problems of vehicle tracking in traffic jams and complex weather conditions such as in sunny, rainy, and cloudy days, or nighttime. In addition, many useful traffic parameters are evaluated from the proposed approach; thus, this traffic behavior analysis may become a reliable means in making an immediate response when abnormal events occur.

This paper is organized as follows. A system overview is provided in Section II. Feature point extraction and tracking are introduced in Sections III and IV. Vehicle behavior analysis is discussed in Section V. Finally, experimental results and conclusions are given in Sections VI and VII.

II. SYSTEM OVERVIEW

Fig. 1 shows the system overview of the proposed approach. There are several steps in the flowchart. First, image pre-processing is used for noise reducing and the elimination of

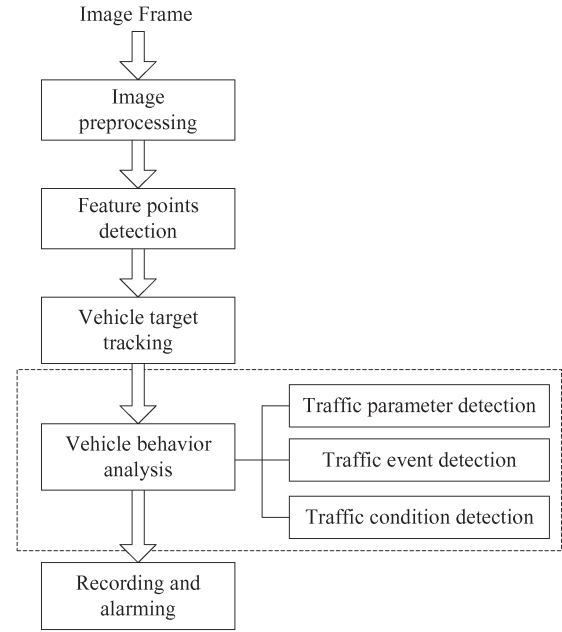


Fig. 1. System overview of the proposed approach.

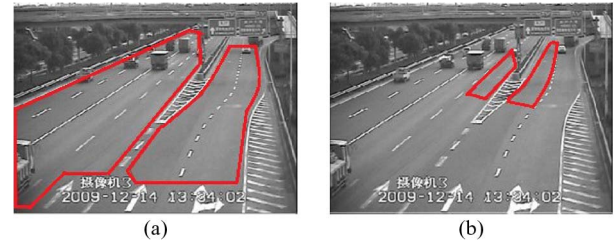


Fig. 2. Configuration of ROI. (a) All lanes as ROI. (b) Retrograde-prone area as ROI.

wobble. Second, feature points of vehicles are extracted from images. Next, feature points are tracked for motion trajectories. Finally, vehicle behaviors are analyzed based on vehicle trajectories, which may provide traffic parameter conditions and event information. To reduce the calculating complexity, all algorithms are applied only in the region of interest (ROI), as shown in Fig. 2. ROI is selected on the upper computer and is optional for addressing users' concerns. For example, users may pay attention to an area where backward running is prone to happen, and then, a specific range may be selected, as shown in Fig. 2(b). In the proposed system, a monocular camera with fixed height and viewing angle is installed to capture the grayscale images.

III. FEATURE POINT EXTRACTION

Vehicle corners commonly appear on high contrast and high curvature points (excluding edges) in video images. Since vehicles are rigid, corner points on a vehicle surface keep their relative positions stable over time, and they can provide coherent motion information of the vehicle. Correspondingly, corner points on the road surface are also detected and tracked, which would be affected by the following vehicle corner tracking. The moving-object extraction would be done before the corner detection to eliminate unnecessary corners and to reduce the calculating amount.

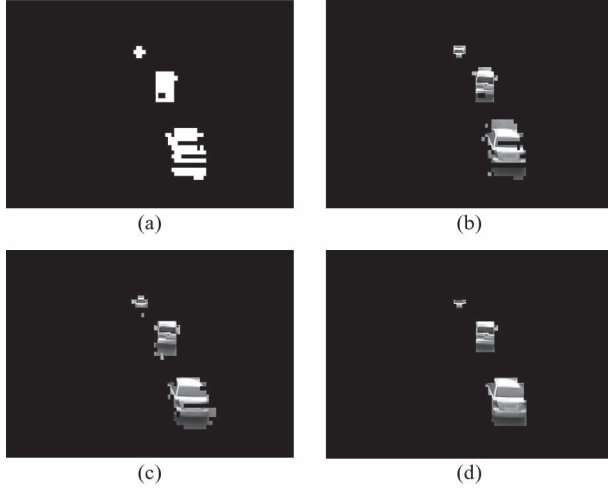


Fig. 3. Moving-object extraction by accumulating frame difference. (a) Subtracted image frame for $t + i$. (b) Subtracted image frame for t . (c) Accumulation of two subtracted images (a) and (b). (d) Result of objects extraction.

A. Moving Object Extraction

We obtain the vehicle candidate through the accumulating frame difference [20], which makes up the incompleteness of the edge obtained by the frame difference. A subtracted image frame is written as $g(t) = f(t) - f(t - i)$. The image frame $h(t)$ accumulating two subtracted images, i.e., $h(t) = |g(t) + g(t - i)|$, is used in the following process, as shown in Fig. 3. The value of i is related to the sampling frequency of video, vehicle speed, and the angle of view. Then, exact vehicle objects are obtained after a morphological opening-and-closing operation.

B. Corner Detection

Corner points are usually detected at occluding contours on the vehicle (two sides of a vehicle or a junction of vehicle windows and body) [7]. However, virtual corner points formed by a vehicle and its background scenes do not physically exist and are extremely unstable during vehicle motion. We use the Moravec detector [5] for detecting corners in the video frames. The Moravec detector calculates the sum of squared differences (SSD) between a detected pixel and its adjacent pixels in the horizontal, vertical, and diagonal directions. We improve the method for reducing noise interference. A 6×8 image block is measured by taking the SSD between the image block and a shifted version of itself. There are eight directions for the block, with one pixel distance away, as shown in Fig. 4. This operation will significantly increase the computation time. Therefore, instead of conducting a SSD, the sum of absolute differences (SAD) is taken between Blocks 1 and 5, 2 and 6, 3 and 7, and 4 and 8, as in (1)–(4), shown below, respectively

$$\text{SAD}_1 = \sum_{i=-2}^3 \sum_{j=-3}^4 |P(x - a + i, y - b + j) - P(x + a + i, y + b + j)| \quad (1)$$

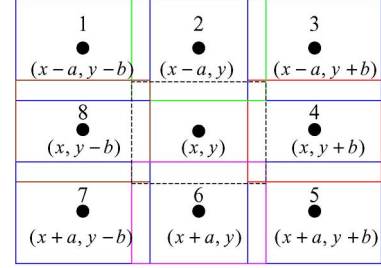


Fig. 4. Improved Moravec detect window.



Fig. 5. Result of corner detection.

$$\text{SAD}_2 = \sum_{i=-2}^3 \sum_{j=-3}^4 |P(x - a + i, y + j) - P(x + a + i, y + j)| \quad (2)$$

$$\text{SAD}_3 = \sum_{i=-2}^3 \sum_{j=-3}^4 |P(x - a + i, y + b + j) - P(x + a + i, y - b + j)| \quad (3)$$

$$\text{SAD}_4 = \sum_{i=-2}^3 \sum_{j=-3}^4 |P(x + i, y - b + j) - P(x + i, y + b + j)| \quad (4)$$

$$\min \text{SAD} = \min\{\text{SAD}_1, \text{SAD}_2, \text{SAD}_3, \text{SAD}_4\} > T. \quad (5)$$

Then, four SAD values will be obtained, and the smallest one will be saved as the eigenvalue $V(x, y)$. If the point eigenvalue is greater than a threshold value T , this point is signed as a corner. Here, $T = M * (6 * 8)$, where $6 * 8$ is the block size, and M is the statistical median of the gray value differences among the pixels of objects and background in the image. Fig. 5 shows the result of corner detection using the aforementioned approach.

C. Corner Selection

We have noticed that several corners may be detected on one vehicle, which may lead to tracking errors and redundant computations. Ideally, only one or a small number of corners should be chosen for a moving object. Since the distribution of corners is usually intensive in a partial region, establishing good or qualified corners to keep their positions stable becomes critical for the following target tracking. The steps for selecting qualified corners are listed in Fig. 6. All the corners are sorted

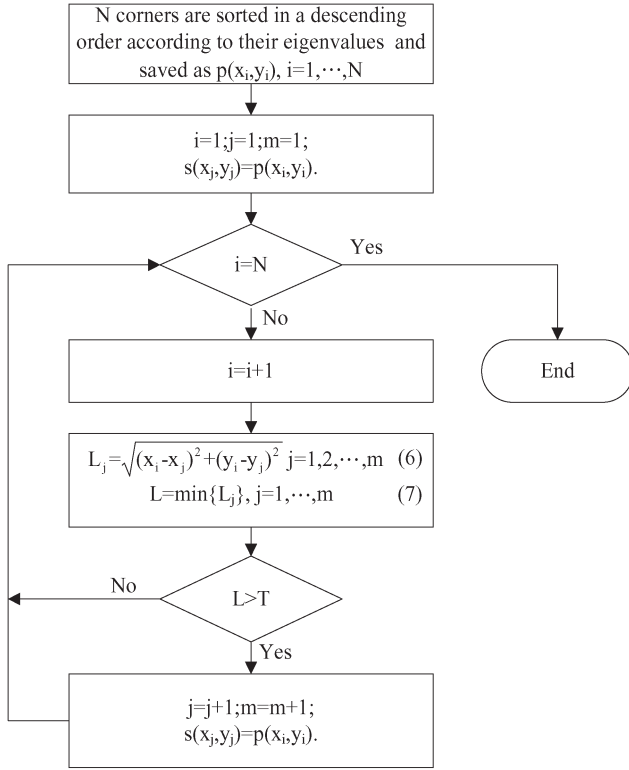


Fig. 6. Procedure of selecting qualified corners.

in descending order according to their eigenvalues. This result will be the scanning order, which is saved as $p(x_i, y_i)$, where i values are from 1 to N , with N being the total number of corners. $s(x_j, y_j)$ is the selected qualified corners, and m is the number of selected corners. As shown in (6) in Fig. 6, L_j is the distance between $p(x_i, y_i)$ and $s(x_j, y_j)$, and L is the shortest distance, which can be derived by checking the minimum L_j for all j in (7) shown in Fig. 6. If L is bigger than threshold T , $p(x_i, y_i)$ is selected as a qualified corner and is saved in $s(x_j, y_j)$, and m increases to $m + 1$. If L is smaller than threshold T , $p(x_i, y_i)$ is not a qualified corner, which indicates that a qualified corner has been selected around $p(x_i, y_i)$. Here, T is the pixels of vehicle in length in the image.

Fig. 7 shows the proposed approach applied to different cases and viewing angles. In Fig. 7(a) and (b), the vehicle corners are selected under the noise-free condition and the noisy condition, respectively. The outdoor road at night and the indoor road with stable light in a tunnel are detected, as shown in Fig. 7(c) and (d). In Fig. 7(e) and (f), highways under snowy and misty conditions are detected.

After selecting corners for the vehicles, tracking is used to measure vehicle paths in video sequence. This operation is performed to provide a correspondence between the regions of consecutive frames based on the corners, which are generated in every video frame. First, considering that vehicle motions that can be treated as uniform linear motions between consecutive frames, vehicle corners will be predicted by Kalman filter [6], which will reduce searching range and calculating amount. Second, a searching area around the predicted corner in the current frame and a matching template area around the corner in the previous frame are established. Then, a full search matching

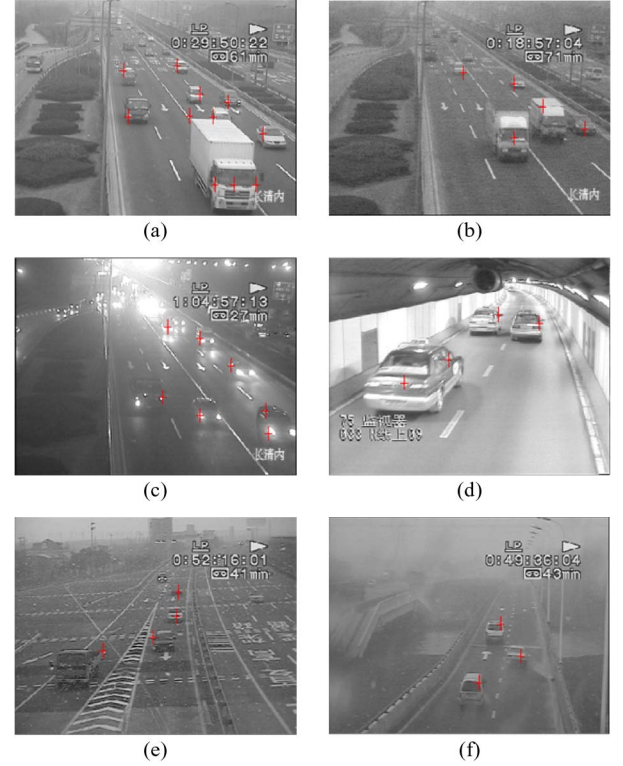


Fig. 7. Corner selection for different scenarios and viewing angles. (a) Under the noise-free condition. (b) Under the noise condition. (c) At night. (d) In a tunnel. (e) Under snowy conditions. (f) Under misty conditions.

is carried out with the searching area and the matching template. Finally, we will get the motion trajectory of every corner [8], whereas irregular track trajectories are removed. The whole procedure is shown in Fig. 8.

It is found that fixed-size searching area and matching template are not suitable for the whole image because vehicle displacement and relative size are different between a close shot and a long shot in the image. Therefore, a dynamic strategy is adopted for size updating of the searching area and matching template. Moreover, the neighborhood pixels of the same corner in every frame are changing; therefore, the matching template needs to be constantly updated. Finally, some irregular track trajectories, which may influence the following vehicle behavior analysis, should be removed. These key methods are introduced in the following.

IV. VEHICLE TRACKING

A. Dynamic Strategy for Size Updating

The ROI is divided into four parts with equal lengths, as shown in Fig. 9(a), and the different sizes of the searching area and matching template, which is decided by the size of each part, is applied in the four parts. With the size of the four parts decreasing, the size of the searching area and matching template will shrink proportionally. For example, if the size of the searching area is defined as $M \times N$ and the size of the matching template is defined as $m \times n$ in Part 1, the size of the searching area will be $M \times N \times CD/AB$, where AB and CD are the widths of Parts 1 and 2, respectively, and the size of the

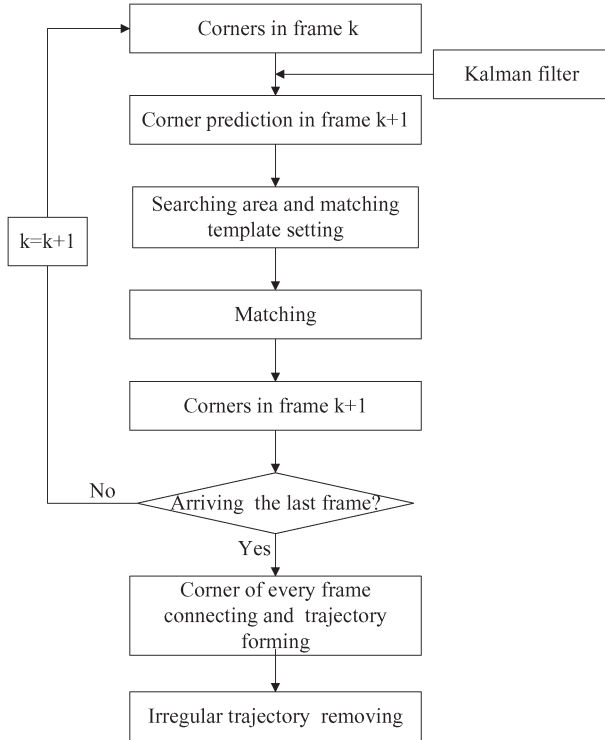


Fig. 8. Procedure of vehicle tracking.

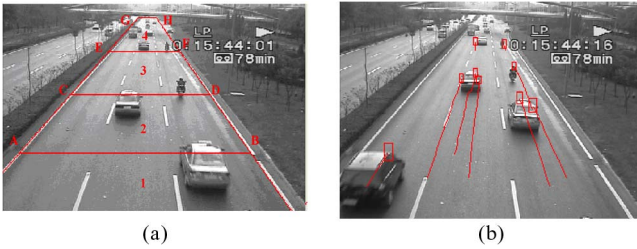


Fig. 9. Searching area and matching template setting. (a) ROI is divided into four parts with equal length. (b) Results of searching for different template.

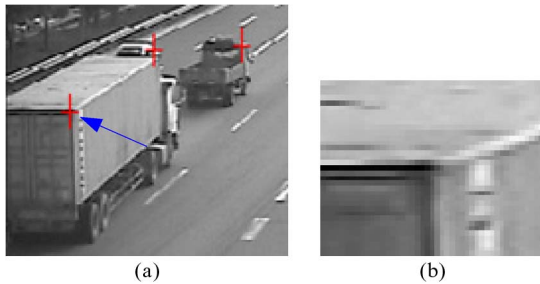


Fig. 10. Matching template. (a) Original image after corners selecting. (b) Matching template is magnified three times.

matching template will be $m \times n \times CD/AB$. The remainder of the parts can be done in the same manner. The results are shown in Fig. 9(b).

B. Matching Template Extraction and Update

A matching template, which is magnified three times in Fig. 10(b), is established centering on the corner, which is indicated by the arrow in Fig. 10(a). We can see that there are

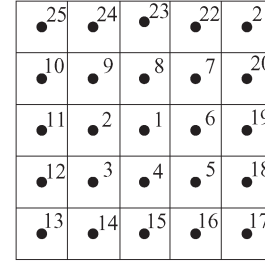
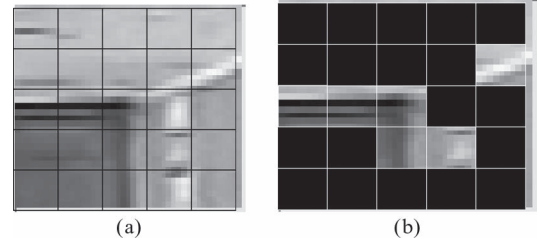
Fig. 11. Matching template is divided into 5×5 blocks.

Fig. 12. Proposed matching template. (a) Matching template mentioned in Fig. 10 is divided into 25 blocks. (b) Results of matching template extraction.

some smooth areas surrounding the corner, which will increase the calculating amount and influence tracking accuracy. To solve the problem, the most similar pixels of the corners are selected for matching. First, the matching template is divided into 5×5 blocks with the same size, as shown in Fig. 11, where Block 1 lies in the center of the matching template, Blocks 2–9 are core blocks, and Blocks 10–25 are surface blocks. Second, every block eigenvalue is calculated according to functions (1)–(5). Next, the matching template consists of Block 1, the three largest eigenvalue blocks in core blocks, and the two largest eigenvalue blocks in surface blocks. Fig. 12 shows the results of a matching template ascertained in accordance with the given methods.

With the movement of vehicles far from the camera, the vehicle image pixels included in the matching template are decreasing; therefore, the structure of the matching template needs to be updated. We have divided the ROI into four parts. When the tracked corners pass across the boundary of two parts, the matching template is updated. In the given method of matching template extraction, six blocks are selected to represent the matching template, which is established in Part 1. With the detected corner entering into Part 2, the block with the smallest eigenvalue in the surface blocks is removed; therefore, there are five blocks representing the matching template. Correspondingly, the number of blocks will decrease by one with the corner crossing the following parts. In Part 4, there are three blocks in the matching template, i.e., Block 1 and the two blocks with the most and second largest eigenvalues in the core blocks.

On the other hand, the neighborhood pixels of the same corner for a vehicle in every frame are changing; therefore, the matching template needs to be constantly updated. The procedure for the matching template updating are shown in Fig. 13. We define two variables i and j , where i is the last time for matching template updating, and j is the current detected corner. If the chessboard distance of $p(x_i, y_i)$ and $p(x_j, y_j)$,

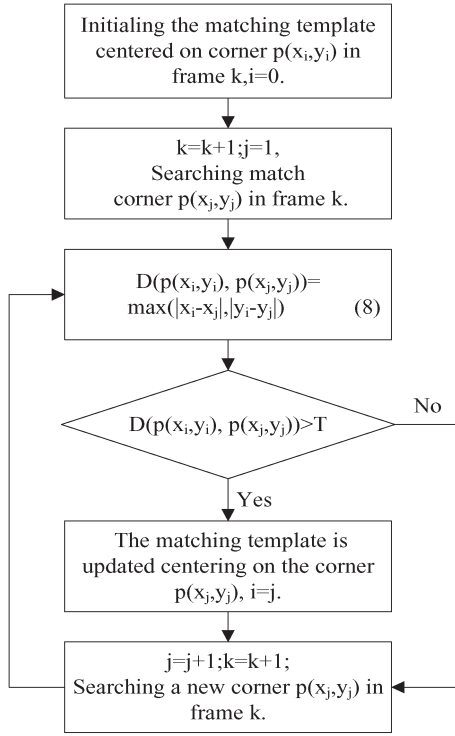


Fig. 13. Procedure for the matching template updating.

as shown in (8) in Fig. 13, is greater than threshold T , the matching template is updated; otherwise, it searches for a new corner in the next frame. Here, The value of T is based on experience with different viewing angles.

C. Irregular Trajectories Removal

A function measuring the degree of similarity between a template and its corresponding portion in the searching area is established. The searching rule is based on

$$\text{SAD}(x, y) = \sum_{i=1}^m \sum_{j=1}^n |f(i+x, j+y) - g(i, j)|$$

$$(x_c, y_c) = \arg(\min_{x, y} (\text{SAD}(x, y))) \quad (9)$$

where (x_c, y_c) is the best predicting position in the searching area, $f(i+x, j+y)$ is the pixel value of the searching area, and $g(i, j)$ is the pixel value of the matching template. Joining up the (x_c, y_c) of every frame, motion trajectories can be completed, as shown in Fig. 14. In Fig. 14(a) and (b), there are two different illumination conditions: one is in the tunnel with a stable light source, and another is at night. Fig. 14(c) shows vehicles moving at high speeds on a snowy day. The misty condition is tested as shown in Fig. 14(d). It indicates that the proposed approach works well under various conditions.

An ideal track trajectory is a uniform change and smooth curve, similar to the trajectories in Fig. 14. However, a few track trajectories are not regular, which may influence the following vehicle behavior analysis. There are two major kinds of irregular track trajectories. When a virtual corner is on an edge of the vehicle and the vehicle color is similar to

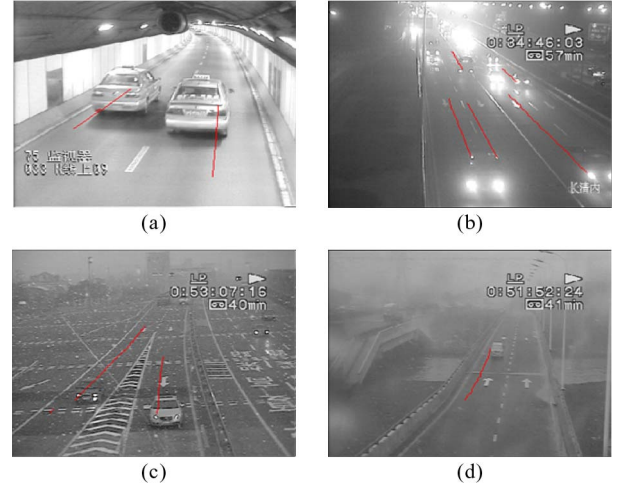


Fig. 14. Testing scenarios under various conditions. (a) In the tunnel with stable light. (b) Nighttime. (c) Snowy day. (d) Misty day.

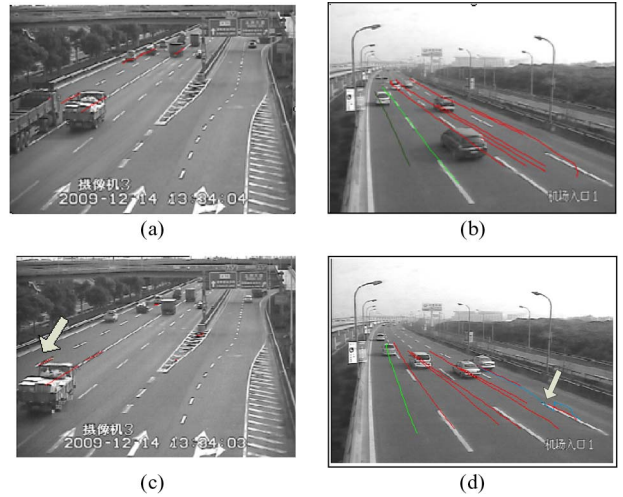


Fig. 15. Irregular track trajectories and results of removing them. (a) Irregular track trajectory formed by mistakenly selected corners. (b) Irregular track trajectory with unsmooth curve. (c) Result of (a) removing the irregular track trajectory. (d) Result of (b) removing the irregular track trajectory.

the background color, the virtual corner is very likely to be mistakenly detected as a vehicle corner, as shown in Fig. 15(a), indicated by the arrow. Another is that the location offsets of detected corners change greatly, and the track trajectory is not smooth, as shown in Fig. 15(b), indicated by the arrow. These irregular track trajectories should be deleted before the vehicle behavior analysis. Therefore, two testing methods are adopted for irregular track trajectories. In the first case, five consecutive frames are selected to calculate the offset of tracked corners in the horizontal and vertical directions. If the offset is smaller than the threshold, remove the track trajectory and stop tracking this corner. In the second case, the arc length and chord length of N consecutive frames are calculated, as follows:

$$\text{Chord} = \sqrt{(x_k - x_{k-N})^2 + (y_k - y_{k-N})^2}$$

$$\text{Arc} = \sum_{i=k-N+1}^k \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (10)$$

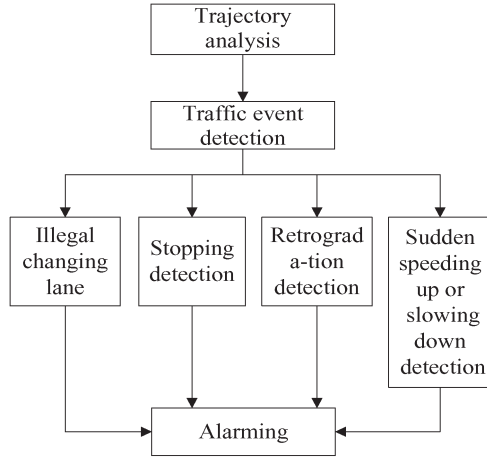


Fig. 16. Vehicle behavior analysis procedure.

where N is the number of consecutive frames, k is the total number of tracked corners, and (x_k, y_k) is the corner coordinate parameter value at k . Then, if the ratio of arc length to chord length is more than the threshold, this track trajectory should be removed. Fig. 15(c) and (d) shows the results of removing irregular track trajectories according to the given two approaches, respectively. These two thresholds are set based on the experience with different viewing angles.

V. VEHICLE BEHAVIOR ANALYSIS

The task of vehicle behavior analysis is to interpret the vehicle behaviors according to vehicle trajectories. Trajectories provide rich information for vehicle motion states, including sudden speeding up and slowing down, stopping, retrogradation (going in a direction not allowed by traffic regulations), lane changing, and other traffic events. All these behaviors can be obtained from **trajectory velocities and directions**. If the **trajectory velocity curve** has an abrupt change, it may indicate that the vehicle is suddenly speeding up or slowing down. If the trajectory direction is opposite from the designed road direction, a retrogradation event may happen. If the trajectory changes greatly in horizontal displacement and exceeds the preset allowable lane width, the vehicle is illegally changing lanes. Fig. 16 shows the vehicle behavior analysis procedure. Once the traffic event is detected, the system will set off an alarm.

A. Speed Estimation

To measure the speeds of vehicles, there must be a mapping from pixels in the image to coordinates in the real world. Although some systems rely on the known height and tilt angle of the camera for calibration [9], lane widths and pavement markings are only used in this approach. The endpoints of pavement markings are labeled with points in the image, as shown in Fig. 17, where the distances among any three points horizontally or vertically are equal. Then, a mapping from pixels in the image to coordinates in the real world can be established based on the known length of pavement markings and lane width.



Fig. 17. Pavement marking calibration.

The instantaneous velocity for traffic event detection is derived from the following:

$$S_i = \frac{T_j - T_k}{F_j - F_k}, \quad k \leq i < j \quad (11)$$

where $F_j - F_k$ is the time difference between frames j and k , and $T_j - T_k$ is the distance between two updated matching templates using the method mentioned in Section IV-B. S_i is the instantaneous velocity of the trajectory in frame i , which is a frame between frame k and frame j .

B. Traffic Event Detection

Traffic events, such as sudden speeding up and slowing down, stopping, retrogradation, lane changing, and other traffic events, may cause serious accidents. Analyzing these vehicle activities can facilitate daily traffic management and immediate responses to abnormal events, consequently providing a more advanced and feasible surveillance scheme. This system is used to detect four types of events, as shown in Fig. 16, and the corresponding methods are introduced in the following. In this part, there are some thresholds for vehicle behavior analysis, and it is related to the installation angle of the camera. Once the camera is installed, the geometric relationship of the video scenes is constructed. Therefore, these thresholds are fixed and are not changed with the same video scenes. These thresholds are set both based on the theories and the experiences in different conditions.

1) *Sudden Speeding Up or Slowing Down*: In general, sudden speeding up or slowing down results in a rapid change of vehicle speeds. Hence, we used the variation rate of vehicle speed (acceleration) for event detection. The instantaneous speed, which is obtained in the given process, is used to calculate the positive or negative accelerations of the vehicles. If the matching template is updated N times, we will get N instantaneous speeds and $N - 1$ accelerations. If the accelerations are continuous positive or negative for αN times, where α values from 0.6 to 1 depend on the requirement of the detection accuracy, the vehicle is speeding up or slowing down.

2) *Stopping*: The vehicle behavior of stopping is different from slowing down. Stopping is usually caused by vehicle breakdowns, crashes, and other active stopping behaviors of drivers. Consequently, the vehicle speed slowly reduces to zero in the stopping process, and the acceleration is negative and



Fig. 18. Calibration of definite road direction. (a) One-way lane calibration. (b) Two-way lanes calibration.

smaller than that in sudden slowing down. The decision rules are listed in the following:

$$V_s < T_{vs} \quad V_e < T_{ve} \quad a_i < T_a \quad (12)$$

where V_s and V_e are the starting speed and ending speed of a trajectory, and a_i is the acceleration of the trajectory in the stopping process. If these three parameters are less than the corresponding thresholds, which are represented by T_{vs} , T_{ve} , and T_a , then a stopping event is happening. The three thresholds are usually set based on experience in several different conditions.

3) *Retrogradation*: First, the calibration of a definite road direction is done before retrogradation detection. We select a lane line as the calibration direction and label it on the upper computer, as shown in Fig. 18. Second, direction vectors of the calibration line are defined as (13) and (14), shown below:

$$\begin{cases} (x_{L1} - x_{L0}, 1), & k = 0 \\ (x_{L(k+1)} - x_{L(k-1)}, 2), & 0 < k < 287 \\ (x_{L287} - x_{L286}, 1), & k = 287 \end{cases} \quad (13)$$

$$\begin{cases} (x_{L0} - x_{L1}, -1), & k = 0 \\ (x_{L(k-1)} - x_{L(k+1)}, -2), & 0 < k < 287 \\ (x_{L286} - x_{L287}, -1), & k = 287 \end{cases} \quad (14)$$

$$\theta = \left| \arccos \left(\frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \sqrt{x_2^2 + y_2^2}} \right) \right|. \quad (15)$$

If the definite direction is far from the camera, the direction vector is represented by (13), where (x_{Lk}, y_{Lk}) is the pixel in the calibration line of Row k . There are 288 rows in the image, and (x_{L0}, y_{L0}) is the pixel of the bottom row in the image. Conversely, if the vehicle is getting close to the camera, the direction vector is represented by (14). Then, the angle θ formed between the definite direction and trajectory direction is expressed as (15), where (x_1, y_1) is the direction vector of the calibration line obtained in the given step, and (x_2, y_2) is the direction vector of the trajectory, which is established by the difference between two points. If θ is larger than 120° for αN times, it indicates that the vehicle is running abnormally, and a high accident probability is expected. α is decided based on experience.

4) *Lane Changing*: Changing lanes is usually forbidden on some road sections. We selected a fixed-lane boundary as the benchmark lane line and labeled it on the upper computer, as



Fig. 19. Detection of illegal lane changing.

shown in Fig. 19. Then, the variance between the benchmark lane line and the trajectory is defined as

$$\text{ave} = \frac{\sum_{i=0}^N (x_{Ti} - x_{Si})}{N} \quad (16)$$

$$D = \frac{\sum_{i=0}^N |x_{Ti} - x_{Si} - \text{ave}|^2}{N} \quad (17)$$

where (x_{Ti}, y_{Ti}) is the pixel of the trajectory in Row i , (x_{Si}, y_{Si}) is the pixel of the benchmark lane line, and N is the number of the tracking frames. The variance shows the diversification of distance between the benchmark lane line and the trajectory. When the vehicle is running normally, the variance has a relatively tiny value. However, when the vehicle changes lanes illegally, the value of the variance increases gradually. When the value of the variance exceeds a threshold that has been set based on experience, it indicates that the vehicle has changed its lane.

VI. EXPERIMENTAL RESULTS

The system test has gone through two major phases. First, we tested a software-only offline system in terms of its ability to detect traffic events. The test provided a “microscopic” view of the system, allowing us to validate the accuracy of the detection results. We prepared many scenarios with different traffic events and in different road sections, including highway and urban sections. These scenarios are tested on a Windows XP platform with a Pentium 4 3.2-GHz central processing unit and 2-GB random access memory. The size of each image is 720×288 , and the sampling frequency is 25 frames/s. The proposed system is implemented with Visual C++ on a raw video format.

Second, the online system was tested on several roads in cities of Xi'an, Shanghai, and Fuzhou in China to see whether the system could accurately detect the traffic events with various conditions accurately, including sunny days, cloudy days, rainy days, and nighttime. We have implemented the surveillance system with a network of cameras that are connected together as shown in Fig. 20. The heavy computational operations of feature point extracting and tracking, and vehicle behavior analysis are placed on the video processors, whereas the detection result and relevant data are saved on the host PC. The video processor is composed of 20 DSPs, and each DSP

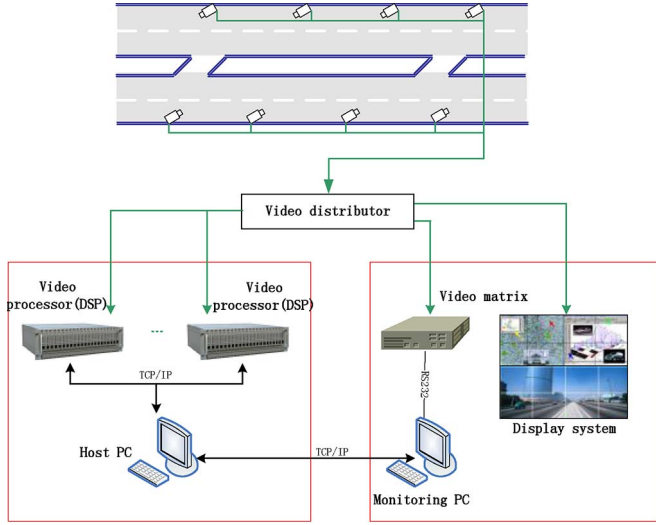


Fig. 20. Online system structure.



Fig. 21. Test scenarios in various conditions. (a) Sunny day. (b) Cloudy day. (c) Nighttime. (d) Rainy day.

processes the data collected by one camera. Meanwhile, the department of traffic management can monitor real-time traffic conditions through the Ethernet.

There are three parts in the experimental results. Analyses for detecting traffic events of software-only systems are addressed in Part A. Part B shows the accuracy ratios and false alarms of traffic event detection with different conditions. Finally, a comparison with other systems is presented in Part C. Test scenarios are shown in Fig. 21. In Fig. 21(a) and (b), there are two different weather conditions: sunny and cloudy. Fig. 21(c) shows vehicles moving at night. The rainy condition is tested in Fig. 21(d). The test conditions for each test scenario are listed in Table I, where H is the installation height of the camera, and θ is the viewing angle. Based on the empirical method, the value of i (see Section III-A) is selected to be fixed to 3 for urban roads and 2 for highways in our test conditions. In Table I, the average processing time per frame with a resolution of 720×288 in various environments is less than 5 ms. In a real-time constraint, the processing time per frame does not exceed 70% of the DSP time. It indicates that the proposed system works well in real time.

TABLE I
CONDITIONS FOR EACH TEST SCENARIO

Scenarios	Place	$H(m)$	θ
Sunny Day	Urban road in Xi'an	12	15
Cloudy Day	Highway in Shanghai	15	15
Night Time	Urban road in Fuzhou	12	15
Rainy Day	Highway in Shanghai	15	15

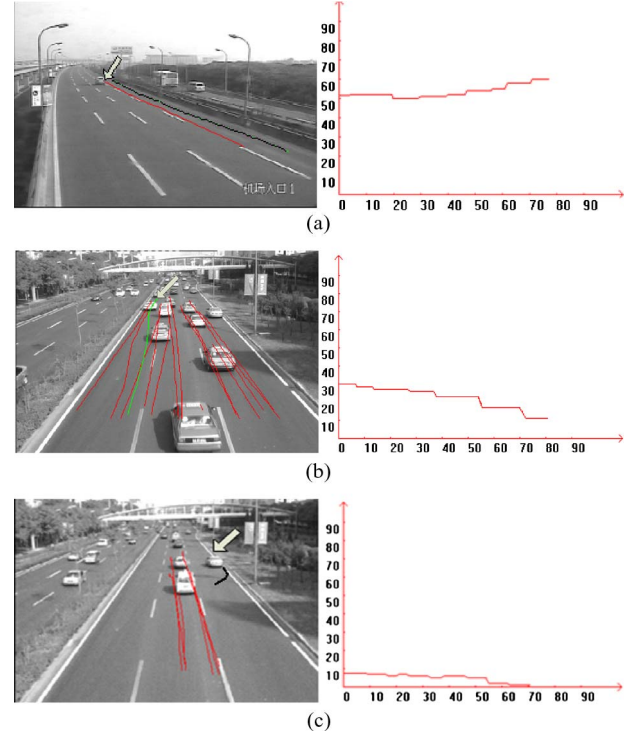


Fig. 22. Instantaneous speed changing curves of different traffic events. (a) Sudden speeding up. (b) Sudden slowing down. (c) Stopping.

A. Analyses of Traffic Event Detection

At the first test phase, we prepared a group of sequences with different events, including sudden speeding up or slowing down and stopping. Then, the changing trends of the trajectories are analyzed. Fig. 22 shows the instantaneous speed changing curves. It can be seen that the instantaneous speed of stopping slowly decreases from about 10 to 0 km/h in 70 frames, as shown in Fig. 22(c), which is different from the case of sudden slowing down. Sudden slowing down is usually caused by abnormal events, such as avoiding pedestrians and traffic accidents; therefore, the vehicle speed will fall down to a lower but not zero speed within a short time. With the decreasing of the vehicle speed, it can be noted that the change frequency of the matching template is declining, as shown in Fig. 22(b). Similarly, Fig. 22(a) shows that the instantaneous speed will increase to a higher value within a short time during the sudden speeding up process. The software-only tests show that the system can detect the traffic events accurately and in a timely manner.




B. Accuracy Ratio and False Alarms of Traffic Event Detection

The second phase of the test focused on three big cities in China with a lot of traffic. During a test period of approximately

TABLE II
TRAFFIC EVENT DETECTION FOR VARIOUS CONDITIONS

Scenarios	Calibration Results					
	Stopping		Retrogradation		Lane changing	
	Detection/ Inspection	Detection rate(%)	Detection/ Inspection	Detection rate(%)	Detection/ Inspection	Detection rate(%)
Sunny Day	732/752	97.3	83/87	95.4	259/272	95.2
Cloudy Day	410/425	96.5	178/189	94.2	162/171	94.7
Rainy Day	549/579	94.8	95/101	94.2	135/144	93.8
Night Time	314/337	93.2	151/164	92.1	73/79	92.4
Scenarios	Testing Results					
	Stopping		Retrogradation		Lane changing	
	Detection/ Inspection	Detection rate(%)	Detection/ Inspection	Detection rate(%)	Detection/ Inspection	Detection rate(%)
Sunny Day	628/650	96.6	97/103	94.2	216/234	92.3
Cloudy Day	560/589	95.1	142/157	90.4	190/207	91.8
Rainy Day	678/722	93.9	116/125	92.8	170/189	90.5
Night Time	391/428	91.4	126/146	88.4	101/114	88.6

TABLE III
EXPERIMENTAL RESULTS OF THE FAR

Scenarios	Calibration Results					
	Stopping		Retrogradation		Lane changing	
	FAC(*)/Inspection	FAR(**) (%)	FAC(*)/Inspection	FAR(**) (%)	FAC(*)/Inspection	FAR(**) (%)
Sunny Day	7/752	0.93	2/87	2.30	10/272	3.68
Cloudy Day	6/425	1.41	5/189	2.65	9/171	5.26
Rainy Day	10/579	1.73	4/101	3.96	8/144	5.56
Night Time	11/337	3.26	6/164	3.66	5/79	6.33
Scenarios	Testing Results					
	Stopping		Retrogradation		Lane changing	
	FAC(*)/Inspection	FAR(**) (%)	FAC(*)/Inspection	FAR(**) (%)	FAC(*)/Inspection	FAR(**) (%)
Sunny Day	10/650	1.54	3/103	2.91	13/234	5.56
Cloudy Day	12/589	2.04	6/157	3.82	15/207	7.25
Rainy Day	22/722	3.05	3/125	2.40	12/189	6.35
Night Time	18/428	4.21	6/146	4.11	10/114	8.77
Fig.23 False alarms of vehicle behavior detection						
	(a) a false alarm of stopping		(b) a false alarm of retrogradation		(c) a false alarm of lane changing	

(*)FAC is the False Alarm Count.

(**)FAR is the False Alarm Ratio, $FAR = FAC / Inspection$.

ten weeks, we studied the data collected under various conditions. The data for each camera are divided into two parts: one is used to calibrate the thresholds, and another is used to evaluate the performance of the system. Because of a lack of accurate visual inspection data for sudden speeding up or slowing down, only stopping, retrogradation, and lane changing were tested. Table II shows the results of traffic event detection in different cases with a calibration data set and a testing data set. We compared the number of the traffic events that were automatically detected (detection) with the number of events detected by human sight for the same sequence (inspection). Based on the results in Table II, we can see that the detection rate at nighttime is the lowest in all cases. From the false detection results, it is noted that the light reflected on the road at night evidently impaired traffic event detection. The proposed system demonstrated good performance on cloudy or rainy days, particularly on highways. Moreover, the detection rates

with the testing data set are slightly lower than the detection rates with the calibration data set, which indicates that the proposed system has good transferability and robustness.

The experimental results of the false alarms (which are also known as false-positive detection) for various scenarios with the calibration data set and the testing data set are listed in Table III. For stopping detection, false alarms seldom appear in the proposed system. However, few false alarms, including discarded objects from vehicles, were misrecognized as vehicle stopping. Fig. 23(a) in Table III includes such an example. In the retrogradation detection, the image jitter is the main cause of false alarms, which will lead to incorrect tracking trajectory. Fig. 23(b) in Table III shows a false alarm caused by an image jitter. When the vehicle is changing lanes, some false alarms were triggered by large trucks, which often appear on the highway. The corner trajectory of a large truck changes unnoticeably; therefore, some lane changing behaviors are missed.

TABLE IV
COMPARISONS WITH OTHER SYSTEMS




System	Manufacturer	Test Sites	Detection Ratio(%)		False Alarm Ratio(%)		Detection Time
			August	September	August	September	
Citilog	Citilog Company of France		88.71%	93.33%	9.74%	7.31%	<10s
Times I	Beijing Aerospace Times Technology Development Co.LTD		85.23%	94.72%	10.56%	5.28%	<10s
The proposed system	The proposed system		90.23%	95.11%	6.13%	4.89%	<5s

Fig. 23(c) shows a case of a large truck causing a false alarm. Based on the experimental results in Table III, the proposed method has low false-alarm counts and false-alarm ratios (FARs) in most scenarios. At night, vehicle-light effects will induce some FARs; therefore, the FAR will be slightly affected. Moreover, the FAR with the testing data set is slightly higher than the FAR with the calibration data set, but it is within an acceptable range in practical applications.

C. Comparison With Other Systems

A comparison with other systems is listed in Table IV. These data are supplied by the company China Shipping Network Technology. Three systems are applied on three urban roads of Shanghai, and the vehicle behaviors, including stopping, retrogradation, and lane changing, are tested in a two-month period, i.e., from August to September. As shown in Table IV, Citilog, which is a famous video-based monitoring and surveillance system made in France, is widely used for road safety around the world. The detection ratio for all the vehicle behavior detection with Citilog is found to be lower than the proposed system. The adaptability of the Citilog monitoring system to the complex traffic environment of China is not high enough, and the foreign system is not updated in time. The detection ratio with Times I is also lower than the proposed system. The proposed system was applied on an elevated freeways of Shanghai, and it can be seen that the proposed system generated better detection results. Meanwhile, the FAR of the proposed system is the lowest in the three systems. From the results in Table IV, we also can see that the detection ratios and FARs in August are lower than those in September because August in Shanghai is a cloudy season, and floating clouds can make the road hazy and bright occasionally, which affects the accuracy of point tracking. In addition, the average detection time is tested in the comparison experiment. In Table IV, the vehicle behavior detection time for Citilog and Times I are less than 10 s, and the proposed system just needs 5 s. The approach using low-level feature tracking makes it possible for the proposed system to process the real-time data in the DSP with video data

analyzed frame by frame. However, in the other two systems, video data are analyzed with frame skipping, which will delay the detection time. Therefore, the proposed system would have better performance in real-time applications.

VII. CONCLUSION

In this paper, a real-time traffic behavior analysis system is proposed. The system is based on the analysis of trajectories provided by the suggested tracking system. Feature points have advantages of efficient locating and easy tracking; therefore, they are used for forming vehicle motion trajectories. The system works well in solving problems of vehicle tracking in complex weather conditions such as sunny, rainy, and cloudy days, as well as nighttime. To reduce the computation time and ensure the real-time performance, an improved Moravec algorithm is used in feature points extraction. Then, trajectories can be obtained by tracking the feature points through the image sequences with a specially designed template, and at the same time, unqualified track trajectories are dropped by decision rules. Based on the experimental results, the tracking accuracy is quite high under complex weather conditions. Traffic events are analyzed according to vehicle trajectories. In offline and online system test phases, the proposed system has not only high detection rates in complex weather conditions but has good time responses in traffic event detection as well. The system has been successfully applied on highways and urban roads of a few big cities in China, and it is found that traffic analysis on highways appears to be less challenging than in the urban environment. In future work, we will improve the detection accuracy for the rain or nighttime condition and enhance the traffic analysis algorithm for the urban environment.

REFERENCES

- [1] G. Yuan, X. Zhang, and Q. M. Yao, "Hierarchical and modular surveillance systems in ITS," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 5, pp. 10–15, Sep./Oct. 2011.
- [2] A. H. Ghods, L. Fu, and A. Rahimi-Kian, "An efficient optimization approach to real-time coordinated and integrated freeway traffic control," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 873–884, Dec. 2010.

- [3] B. F. Wu, C. C. Kao, and C. C. Liu, "The vision-based vehicle detection and incident detection system in Hsueh-Shan Tunnel," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2008, vol. 8, pp. 1394–1399.
- [4] Y. K. Ki and D. Y. Lee, "A traffic accident recording and reporting model at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 188–194, Jun. 2007.
- [5] H. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-80-03, 1980.
- [6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng. D*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
- [7] D. Bloisi and L. Iocchi, "Argos—A video surveillance system for boat traffic monitoring in Venice," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 7, pp. 1477–1502, Nov. 2009.
- [8] B. Johansson, J. Wiklund, P. Forssen, and G. Granlund, "Combining shadow detection and simulation for estimation of vehicle size and position," *Pattern Recognit. Lett.*, vol. 30, no. 8, pp. 751–759, Jun. 2009.
- [9] E. K. Bas and J. D. Crisman, "An easy to install camera calibration for traffic monitoring," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 1997, pp. 362–366.
- [10] S. Kamijo and H. Inoue, "Incident detection from low-angle images of heavy traffics in tunnels," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2007, pp. 81–86.
- [11] H. Ikeda and T. Mastuo, "Abnormal incident detection system employing image processing technology," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 1999, pp. 748–752.
- [12] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011.
- [13] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Juan, Puerto Rico, Jun. 1997, pp. 495–501.
- [14] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–151.
- [15] C. P. Lin, J. C. Tai, and K. T. Song, "Traffic monitoring based on real-time image tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, vol. 2, pp. 2091–2096.
- [16] N. K. Kanhere and S. T. Birchfield, "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 148–160, Mar. 2008.
- [17] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, "Detection and classification of highway lanes using vehicle motion trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 188–200, Jun. 2006.
- [18] Y. Wang, "Joint random field model for all-weather moving vehicle detection," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2491–2501, Sep. 2010.
- [19] J. M. A. Alvarez and A. M. Lopez, "Road detection based on illuminant invariance," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 184–193, Mar. 2011.
- [20] J. Zhou, D. Gao, and D. Zhang, "Moving vehicle detection for automatic traffic monitoring," *IEEE Trans. Veh. Technol.*, vol. 56, no. 1, pp. 51–59, Jan. 2007.
- [21] W. M. Hu, X. J. Xiao, D. Xie, T. N. Tan, and S. Maybank, "Traffic accident prediction using 3D model based vehicle tracking," *IEEE Trans. Veh. Technol.*, vol. 53, no. 3, pp. 677–694, May 2004.
- [22] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 108–118, Jun. 2000.



Sheng-Nan Lu was born in Jiangsu, China, in 1982. She received the B.S. and M.S. degrees in computer science and technology from Chang'an University, Xi'an, China, in 2004 and 2007, respectively. She is currently working toward the Ph.D. degree in intelligent transportation and information system engineering with Chang'an University.

She is currently a Lecturer with the School of Computer, Xi'an Shiyou University, Xi'an. Her research interests include image processing in intelligent transportation systems and pattern recognition.



Xiang Ma received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2011.

From 2010 to 2011, he was with the Image Processing and Interpretation Laboratory, Ghent University, Ghent, Belgium, as an exchange Ph.D. student, supported by the European Union. He is currently a Lecturer with the School of Information Engineering, Chang'an University, Xi'an. He is the author of over 20 referenced journal and conference papers. His research has been supported by the National Natural Science Foundation of China. His research

interests include facial image processing and image processing in intelligent transportation.



Yuan Yang was born in Shanxi, China, in 1987. She received the B.S. degree in communication engineering from Chang'an University, Xi'an, China, in 2011. She is currently working toward the M.S. degree with the Department of Information Engineering, Chang'an University.

Her research interests include computer vision, image processing, and intelligent transportation systems.



Xue-Qin Liu was born in Hebei, China, in 1988. She received the B.S. degree in communication engineering from Chang'an University, Xi'an, China, in 2011. She is currently working toward the M.S. degree in computer software with Chang'an University.

Her research interests include image processing in intelligent transportation systems, pattern recognition, and computer vision.



Huan-Sheng Song was born in Inner Mongolia, China, in 1964. He received the B.S. and M.S. degrees in communication and electronic systems and the Ph.D. degree in information and communication engineering from Xi'an Jiaotong University, Xi'an, China, in 1985, 1988, and 1996, respectively.

Since 2004, he has been with the Information Engineering Institute, Chang'an University, Xi'an, where he became a Professor in 2006 and was nominated as the Dean in 2012. He has been involved

in research on intelligent transportation systems for many years and has led a research team to develop a vehicle license plate reader and a traffic event detection system based on video, which has brought about complete industrialization. His current research interests include image processing and recognition, as well as intelligent transportation systems.



Peng Zhang was born in Hohhot, China, in 1982. He received the B.S. and M.S. degrees in computer science from the University of Science and Technology of China, Hefei, China.

He is currently with China Highway Engineering Consulting Group Company Ltd., Beijing, China. His research interests include intelligent transport systems and wireless communication networks.