Adaptive Vehicle Detector Approach for Complex Environments

Bing-Fei Wu, *Fellow, IEEE*, and Jhy-Hong Juang

***Abstract*—In this paper, a vehicle detection approach for complex environments is presented. This paper proposes methods for solving problems of vehicle detection in traffic jams and complex weather conditions such as sunny days, rainy days, cloudy days, sunrise time, sunset time, or nighttime. In recent research, there have been many well-known vehicle detectors that utilize background extraction methods to recognize vehicles. In these studies, the background image needs to continuously be updated; otherwise, the luminance variation will impact the detection quality. The vehicle detection under various environments will have many difficulties such as illumination vibrations, shadow effects, and vehicle overlapping problems that appear in traffic jams. The main contribution of this paper is to propose an adaptive vehicle detection approach in complex environments to directly detect vehicles without extracting and updating a reference background image in complex environments. In the proposed approach, histogram extension addresses the removal of the effects of weather and light impact. The gray-level differential value method is utilized to directly extract moving objects from the images. Finally, tracking and error compensation are applied to refine the target tracking quality. In addition, many useful traffic parameters are evaluated. These useful traffic parameters, including traffic flows, velocity, and vehicle classifications, can help to control traffic and provide drivers with good guidance. Experimental results show that the proposed methods are robust, accurate, and powerful enough to overcome complex weather conditions and traffic jams.**

***Index Terms*—Histogram extension (HE), tracking compensation, tracking, traffic jam, vehicle detection.**

I. Introduction

WITH the growing number of vehicles, traffic information increasingly becomes important for drivers. Many approaches have been proposed for tackling related problems in intelligent transportation system (ITS). Wang [1] proposed a joint random field (JRF) model for moving vehicle detection in video sequences. The proposed method could handle moving cast shadows, lights, and various weather conditions. However, the method did not recognize vehicle classification and velocity. Tsai *et al.* [2] presented a novel vehicle detection approach for detecting vehicles from static images using color and edges. This method introduced a new color transform model to find important "vehicle color" for quickly locating possible vehicle candidates. This method could also detect vehicles in various weather conditions, but it did not address resolutions on traffic jams and shadow reduction. Zhang *et al.* [3] developed a multilevel framework to detect and handle vehicle occlusion. The proposed framework consisted of intraframe, interframe, and tracking levels to resolve the occluded vehicles. Neeraj *et al.* [4] gave a method for segmenting and tracking vehicles on highways using a camera that was relatively low. Melo [5] described a low-level object tracking system that produced accurate vehicle motion trajectories, which could further be analyzed to detect lane centers and classify lane types. A lane-detection method that was aimed at handling moving vehicles in traffic scenes was proposed by Cheng *et al.* [6]. A new background subtraction algorithm based on the sigma–delta filter, which was intended to be used in urban traffic scenes, was presented in [7]. An example-based algorithm for moving vehicle detection was introduced in [8]. In addition, many approaches have been proposed for tackling related problems in ITS. The model-based approach [9] uses a 3-D model to detect vehicles. In this method, different models that correspond to different types of vehicles are created. Song *et al.* [10] and Koller *et al.* [11] used an active contour method to track vehicles. In this method, the vehicles could easily be tracked, and computation loading could significantly be reduced. However, system initialization was a critical risk. Coifman [12] developed a vision-based system with gradient operator to detect subcorner features of the vehicles and grouped these features to detect the vehicles. The advantage of this method was that it was less sensitive to change in illumination. On the other hand, this method could meet the challenge of determining grouping conditions. Wang *et al.* [13], detected the motion information of the spatial–temporal wavelet of video sequence. Cucchiara *et al.* [14] integrated moving edge detection and headlight detection into a hybrid system. This system worked not only during the day but also at night. Unlike most methods referring to background image, they used a three-image difference to detect moving edge. This method reduced both the dependence on background and the time of background learning. However, noise affected the system to a great extent. Background segmentation was one approach for extracting the common part between different images in a frame. With good flows of learning and updating, objects could more completely be extracted. Beymer *et al.* [15] proposed a vehicle-tracking algorithm to estimate traffic parameters using corner features. In addition, Liao *et al.* [16] used entropy as an underlying measurement to calculate traffic flows and vehicle speeds. Baker *et al.* [17] proposed a 3-D model-matching scheme to classify vehicles into various types, such as wagons, sedan, and hatchback. Furthermore, Gupte *et al.* [18]

Manuscript received September 2, 2010; revised April 21, 2011 and August 9, 2011; accepted December 3, 2011. Date of publication February 3, 2012; date of current version May 30, 2012. This work was supported in part by the National Science Council, Taiwan, under Contract NSC 100-2221-E-009-041. The Associate Editor for this paper was A. Amditis.

The authors are with the Institute of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan (e-mail: jackie_juang@springsoft.com; bwu@cc.nctu.edu.tw).

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/TITS.2011.2181366

proposed a region-based approach to track and classify vehicles based on the establishment of correspondences between regions and vehicles. In [15], [18], and [19], a manual method of camera calibration has been presented to identify lane locations and corresponding lane widths.

Finally, many recent ITS-related studies have been proposed. Bartin [20] proposed a case study where the objective was to identify the optimal subset of routes for real-time traveler information in a highway network. Xie [21] developed a privacy-aware monitoring system (PAMS) that worked as an aggregate query processor that protected the location privacy of drivers as it made the IDs of cars anonymous. Chang [22] presented a regulator to track the optimal vehicle-detector location in a variety of traffic conditions and an algorithm to adjust the detected data from the original fixed detector as if they were detected by the detector at its time-dependent optimal location. Randeniya *et al.* [23] presented the results of an effort where position and orientation data from vision and inertial sensors were integrated and validated using data from an actual roadway. Kanhere [24] presented a taxonomy for roadside camera calibration that not only encompassed the existing methods [two vanishing points and known width (VVW), one vanishing point, known width and camera height (VWH), and one vanishing point, known width and length (VWL)] but included several novel methods [two vanishing points and known camera height (VVH), two vanishing points and known length (VVL), one vanishing point, known length and camera height (VLH), two vanishing points, known distance to the road (VVD), one vanishing point, known width, and distance (VWD), and one vanishing point, known height, and distance (VHD)] as well. Wang [25] proposed an overview of the background, concepts, basic methods, major issues, and current applications of parallel transportation management systems (PtMS). Liu [26] introduced a queuing network-based computational model to explain driver performance in a pedestrian-detection task assisted with night vision enhancement systems. Ghods [27] tackled the problem of real-time optimal control of traffic flow in a freeway network deployed with coordinated and integrated traffic controllers. Alvarez *et al.* [28] proposed a novel approach to vision-based road detection that was robust to shadows.

Although there were many studies on vehicle detection, few research proposed methods for solving problems of detecting vehicles in various complex environments, particularly in traffic jams that frequently occur in practical traffic conditions. The vehicle detection under various environments will meet many difficulties such as illumination vibrations, shadow effects and vehicle overlapping problems that appear in traffic jams. In this paper, an adaptive vehicle detection approach for complex environments is proposed for solving problems of vehicle detection in traffic jams and complex weather conditions like sunny days, rainy days, sunrise, sunset, cloudy days, fog, or at night. Histogram extension (HE) addresses how we can remove effects of weather and light impact. The gray-level differential value method (GDVM) is used to dynamically segment moving objects. Finally, tracking and error compensation are applied to refine the target tracking quality. In addition, many useful traffic parameters are evaluated from the proposed approach, including traffic flows, velocity, and vehicle classifications. These useful parameters can help in controlling traffic and provide drivers good driving guidance.

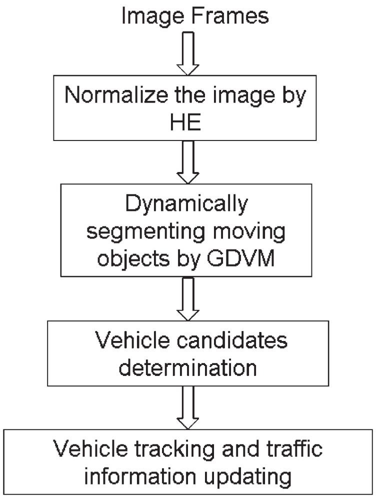
Image Frames
Normalize the image by HE
Dynamically segmenting moving objects by GDVM
Vehicle candidates determination
Vehicle tracking and traffic information updating

Fig. 1. System overview of the proposed approach.

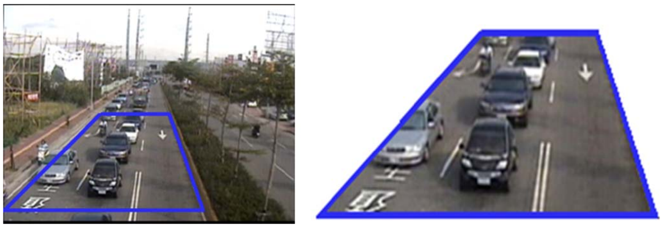

Fig. 2. Source image and its ROI.

This paper is organized as follows. Section II addresses the system overview. Section III shows the HE method, which makes images from various environments similar to each other. Dynamic moving object segments, including GDVM and the tracking procedure, are presented in Section IV. Finally, experimental results are given in Section V, and the conclusion is presented in Section VI.

II. System Overview

Fig. 1 shows the system overview of the proposed approach. There are several steps in the flowchart. First, images are normalized by HE. Second, moving objects are dynamically segmented by GDVM. Next, vehicle candidate detection, vehicle tracking, and error compensation are applied. Finally, the traffic parameters are evaluated and updated. To reduce computing loads, all algorithms are applied only in the region of interest (ROI), as shown in Fig. 2. In the proposed system, a monocamera is installed to capture the full color images. The viewing angle is the front view. The camera has fixed height and viewing angle.

III. Histogram Extension

Input image frames in various environments have various properties. These fluttering properties may damage the detection quality. In practical conditions, a vehicle detection system must work well in all kinds of complex environments. Different

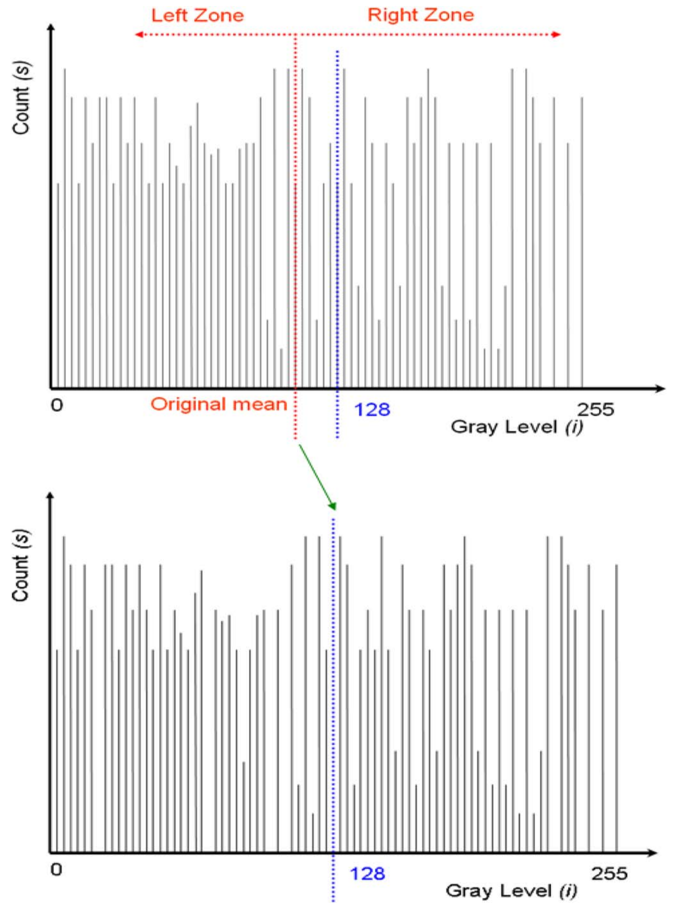

Fig. 3. LNMS.

environments produce different light effects that make vehicle detection hard to work well. HE is a method that removes the effects of weather and light impact.

The first step for the HE method is to decompose a true-color image into its red–green–blue (RGB) components and calculate the histogram in the ROI for each component. Next, linear normalization with mean shift (LNMS) is applied to normalize the source images. There are several steps for the LNMS method. First, the original mean value of all pixels, denoted as m_p in (1), shown below, is shifted to 128. Next, α and β, which are defined as shifting parameters in (2), shown below, are calculated for the left and right zones, respectively, as shown in Fig. 3. Finally, all gray levels that correspond to gray-level counts, denoted as $i_L(s)$ in the left zone and $i_R(s)$ in the right zone, are normalized to the new ones, denoted as $ni_L(s)$ and $ni_R(s)$. In (3), shown below, $i_L(s)$ is the original gray level that corresponds to the gray-level count in the left zone, $i_R(s)$ is the original gray level that corresponds to the gray-level count in the right zone, $ni_L(s)$ is the shifted gray level that corresponds to the gray-level count in the left zone, and $ni_R(s)$ is the shifted gray level that corresponds to the gray-level count in the right zone. We have

$$m_p = \frac{\sum_{i=0}^{255} i \times s(i)}{\sum_{i=0}^{255} s(i)} \tag{1}$$

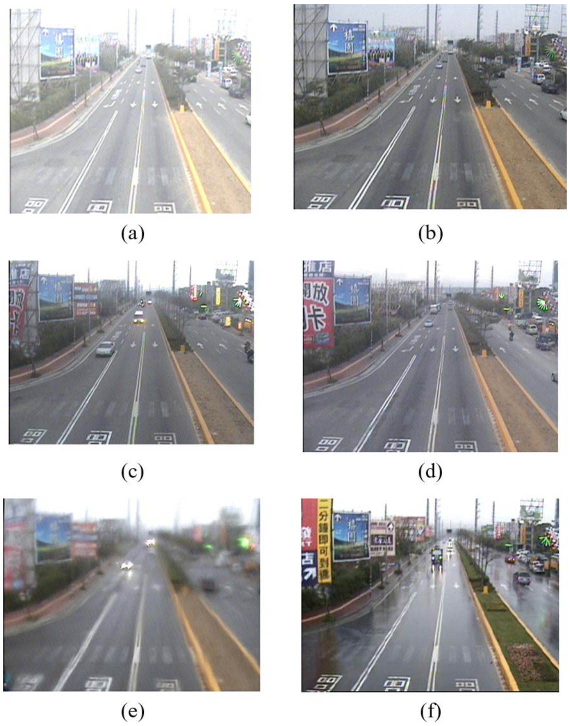
(a) (b) (c) (d) (e) (f)

Fig. 4. Experiment scenarios for different weather conditions. (a) Sunny. (b) Daytime. (c) Evening. (d) Cloudy. (e) Blurred. (f) Rainy.

where i is the gray-level value, and $s(i)$ is the pixel count in gray level i. In addition

$$\alpha = \frac{128}{m_p}$$
$$\beta = \frac{128}{255 - m_p} \tag{2}$$
$$ni_L(s) = i_L(s) \times \alpha$$
$$ni_R(s) = i_R(s) \times \beta. \tag{3}$$

When LNMS is applied to each component of RGB, the new mean values of R, G, and B will approach 128. In addition, the gray-level scale in the left zone is smoothly normalized to 0 and 128, and the gray-level scale in the right zone is smoothly normalized to 128 and 255.

The experimental cases for proving the performance of LNMS are shown in Fig. 4. There are six test sceneries, including sunny, daytime, evening, cloudy, blurred, and rainy. The root mean square error (RMSE) in (4), shown below is utilized to compare the differences between two images in different testing scenarios. In (4), E_f is defined as the RMSE of two images f_1 and f_2. $f_1(x,y)$ and $f_2(x,y)$ are defined as the pixel value at (x,y) in f_1 and f_2

$$E_f = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f_1(x,y) - f_2(x,y)\right]^2} \tag{4}$$

where M is the height of the image, and N is the width of the image. Table I lists the RMSE comparisons among different scenarios before applying HE, and Table II lists the comparison of images after applying HE. Based on the experimental results,

TABLE I
COMPARISON OF THE RMSE OF SOURCE IMAGES AMONG VARIOUS WEATHER CONDITIONS (LUMINANCE Y)

	Sunny	Daytime	Evening	Cloudy	Blurred	Rainy
Sunny	0	37.39	41.34	38.78	32.26	37.38
Daytime		0	32.98	25.32	27.53	38.41
Evening			0	24.39	20.54	36.59
Cloudy				0	19.31	32.45
Blurred					0	24.29
Rainy						0

TABLE II
COMPARISON OF THE RMSE OF IMAGES APPLYING HE AMONG VARIOUS WEATHER CONDITIONS (LUMINANCE Y)

	Sunny	Daytime	Evening	Cloudy	Blurred	Rainy
Sunny	0	15.49	21.66	17.12	25.3	23.54
Daytime		0	15.06	14.28	21.49	26.83
Evening			0	11	15.36	17.46
Cloudy				0	17.89	20.81
Blurred					0	18.11
Rainy						0

all source images can significantly reduce the light effects. This benefit simplifies system-parameter setting, noise reduction, and moving object segments. In addition, HE makes the system more robust to overcome problems in complex environments.

IV. DYNAMIC MOVING OBJECT SEGMENT AND TRACKING PROCEDURE

There are several steps to track vehicles in the proposed vehicle detector. First, GDVM is used to dynamically segment moving objects. Second, vehicle candidates can be extracted from the moving objects by merging fractal objects or splitting mismerged objects. Next, a tracking procedure is used to guarantee the detection quality, including filtering noises. Finally, to improve the accuracy of traffic parameters and ensure the stability of the tracking flow, a tracking compensation method is also proposed. In a vehicle detector, the desired moving objects should be segmented from the road surface. To segment the correct moving objects fast without using background concepts, GDVM among the R, G, and B components is applied in the proposed system.

A. Dynamically Segmenting Moving Objects

GDVM is used to segment moving objects from the background. Gray road surface and white or yellow lane marks are assumptions in GDVM. The remaining colors are taken as moving objects on the road. For gray, white, and yellow, ΔRG, ΔRB, and ΔGB are small in (5), shown below. In practical cases, most nongray cars, including white and black cars, can be segmented by (6), shown below. To extract gray-like cars, because the luminance Y of white cars is higher and the Y of dark cars is lower than the road surface, the Y value of the road surface always locates by excluding the range between the two threshold values. The green component G of the RGB model contributes around 60% to Y. Therefore, the G value can be adopted to reduce the computation loading, and gray-like cars can be segmented by compensating for the moving objects. In (6), a true color image can be transformed to a binary moving object at (x, y), which is denoted as $MO(x, y)$

$$\begin{aligned}\Delta RG(x,y) &= |R(x,y) - G(x,y)|\\ \Delta RB(x,y) &= |R(x,y) - B(x,y)|\\ \Delta GB(x,y) &= |G(x,y) - B(x,y)| \end{aligned} \tag{5}$$

$$MO(x,y) = \begin{cases} 1 & \begin{array}{l}\Delta RG(x,y) > TH_{\mathrm{RG}} \text{ and}\\ \Delta \mathrm{RB}(x,y) > TH_{\mathrm{RB}} \text{ and}\\ \Delta \mathrm{GB}(x,y) > TH_{\mathrm{GB}} \text{ or}\\ (TH_{low} \geq \mathrm{G}(x,y) \text{ or}\\ \mathrm{G}(x,y) \geq TH_{\mathrm{high}})\end{array}\\ 0 & \text{otherwise.}\end{cases} \tag{6}$$

In (6), TH_{RG}, TH_{RB}, and TH_{GB} are thresholds for ΔRG, ΔRB, and ΔGB. TH_{low} and TH_{high} are thresholds for $G(x, y)$ The threshold values in (6) are determined by applying an adaptive thresholding procedure. We take the calculation of TH_{RG} as an example. In (7), shown below, $f_{RG}(x, y, n)$ is assigned to 1 when $\Delta RG(x, y) = n$. Next, the filtered difference distribution $FD_{RG}(n)$ in (9), shown below, can be derived by calculating $D_{RG}(n)$, which is a difference distribution, as in (8), shown below. In (9), $2p + 1$ is the filter order of the specified moving average filter. Finally, TH_{RG} in (11), shown below, can be obtained by calculating the Laplacian operator in (10), shown below:

$$f_{RG}(x,y,n) = \begin{cases} 1, & \Delta RG(x,y) = n\\ 0, & \text{otherwise}\end{cases} \tag{7}$$

$$D_{RG}(n) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f_{RG}(x,y,n) \tag{8}$$

$$FD_{RG}(n) = \frac{\sum\limits_{i=n-p}^{n+p} D_{RG}(i)}{2p+1} \tag{9}$$

$$\begin{aligned}\nabla^2 FD_{RG}(n) = {} & FD_{RG}(n+1) - 2FD_{RG}(n)\\ & + FD_{RG}(n-1)\end{aligned} \tag{10}$$

$$TH_{RG} = \min_n \left(\arg\left(\nabla^2 FD_{RG}(n) = 0\right)\right). \tag{11}$$

The same calculations are applied to obtain TH_{RB}, TH_{GB}, TH_{low}, and TH_{high}, as in (12), shown below, respectively

$$\begin{aligned} TH_{RB} &= \min_n \left(\arg\left(\nabla^2 FD_{RB}(n) = 0\right)\right)\\ TH_{GB} &= \min_n \left(\arg\left(\nabla^2 FD_{GB}(n) = 0\right)\right)\\ TH_{low} &= \min_n \left(\arg\left(\nabla^2 FD_{G}(n) = 0\right)\right)\\ TH_{high} &= \max_n \left(\arg\left(\nabla^2 FD_{G}(n) = 0\right)\right). \end{aligned} \tag{12}$$

Finally, the example for applying GDVM is illustrated in Fig. 5.

B. Detect Vehicle Candidates by Merging and Splitting Moving Objects

In practical conditions, a vehicle candidate may be broken into several moving objects in the $MO(x, y)$ domain. On the other hand, two or more closing vehicles may incorrectly be detected as one moving object. Methods of merging and splitting moving objects should be applied to more precisely detect

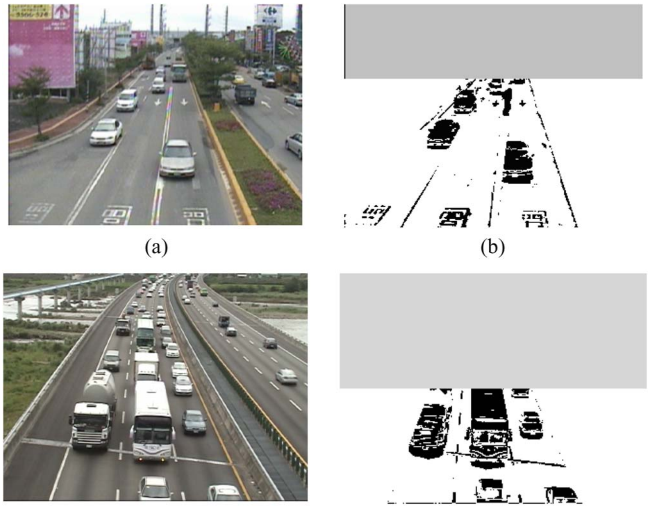

Fig. 5. Examples for applying GDVM. (a) Original image in common traffic. (b) Segmented result. (c) Original image in jams. (d) Segmented result.

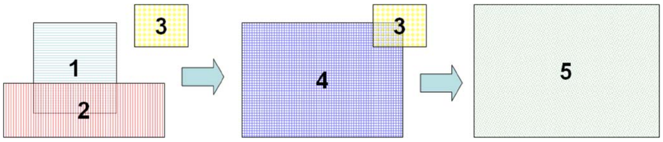

Fig. 6. Example of MBBR by merging RBBs.

vehicle candidates. There are several steps for these methods. First, the merge boundary box rule (MBBR) is applied to merge the moving objects. The moving objects may be detected as many small rectangle boundary boxes (RBBs) in $MO(x, y)$. MBBR is a method of merging the overlapped RBBs into a large box. When two RBBs overlap, a new RBB is rebuilt to combine and replace the two old RBBs. In Fig. 6, when two RBBs, which are denoted as RBB1 and RBB2, overlap, they are merged into a new RBB, RBB4. In a recursive way, RBB5 is created by merging RBB3 and RBB4. Finally, MBBR is stopped when all overlapped RBBs are merged.

After applying MBBR, fractal moving objects are merged as more solid ones, as shown in Fig. 7. There are several attributes for determining a moving object as a vehicle candidate, including the width, height, width/height ratio, and density of the moving object. When moving objects have suitable attributes, they are identified as vehicle candidates. Otherwise, they are further merged or split. If the attributes of moving objects exceed the limit and cannot be merged or split, they are filtered as noise. These attributes of moving objects are influenced by various practical testing environments, including camera settings, image resolutions, and viewing angles. The testing conditions for various environments are listed in Table IV.

When a vehicle is broken into several moving objects, some conditions should be met. First, the adjacent moving objects should have similar width and density. Second, the two moving objects should be shown as close. Next, the new merged moving object should have suitable attributes, including width, height, and density. Once they meet these conditions, the moving objects should be merged as a new moving object.

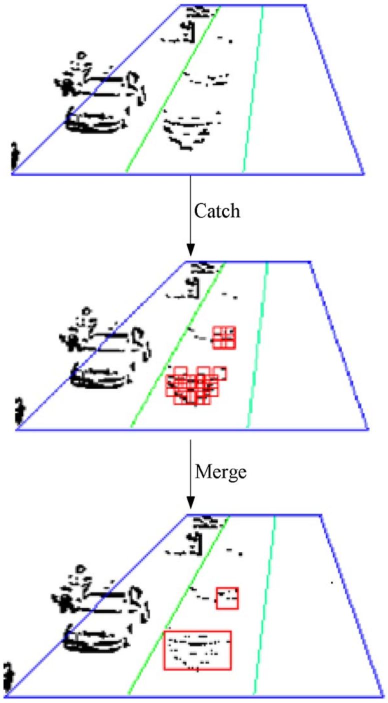

Fig. 7. Example of merging fractal moving objects by MBBR.

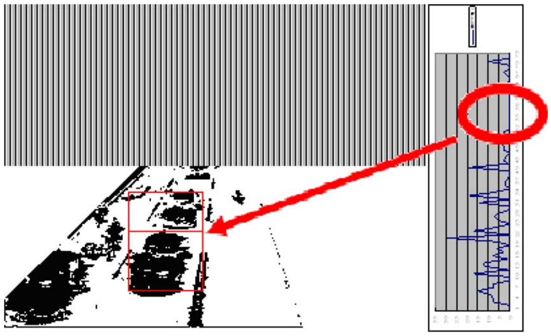

Fig. 8. Vertical projection gap between closing vehicles.

When two vehicles are too close and are mismerged as one moving object, they should be split into two or more moving objects. The steps for detecting and resolving the mismerged moving objects are listed as follows:

Step 1. Check if the moving object is a mismerged moving object. A mismerged moving object has improperly large height H, width W, H/W ratio, and density. If it is a mismerged moving object, go to step 2. If it is not a mismerged moving object, the step is terminated for the moving object.

Step 2. Find the gap in the vertical histogram projection of the moving object in Fig. 8. First, a low-pass filter (13), shown below, is applied to the vertical histogram projection, where $vp(n)$ is the histogram value at n, and $vp^*(n)$ is the filtered histogram value. Second, a sliding window is used to gain the sum of the vertical histogram $svp(n)$ at n with $2M + 1$ points in (14), shown below. Next, $ssvp(n)$ in (15), shown below, which is also similar to a sliding window, is calculated based on $svp(n)$ at n. The gap position, which is denoted as n_{gap}, can be derived by checking the

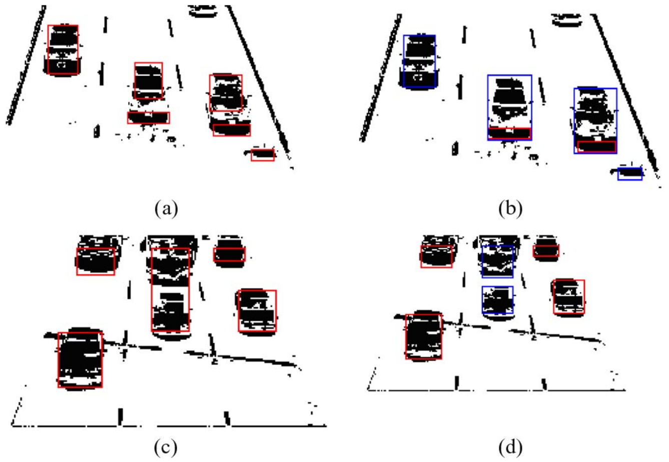

Fig. 9. Examples of merging and splitting moving objects. (a) Before merging moving objects. (b) After merging moving objects. (c) Before splitting moving objects. (d) After splitting moving objects.

minimum $ssvp(n)$ for all n in (16), shown below. Finally, the mismerged moving object is split by the gap as

$$vp^*(n) = \frac{\sum_{i=-N}^{N} vp(n+i)}{2N+1} \tag{13}$$

$$svp(n) = \sum_{i=-M}^{M} vp^*(n+i) \tag{14}$$

$$ssvp(n) = \sum_{i=-X}^{X} svp(n+i) \tag{15}$$

$$n_{gap} = \arg_n (\min (ssvp(n))) \tag{16}$$

where $2N+1$, $2M+1$, and $2X+1$ are the filter orders.

Step 3. All moving objects and the new split moving objects should repeatedly be checked from step 1 until all moving objects are evaluated.

Fig. 9 shows an example of merging and splitting moving objects. In Fig. 9(a), the vehicle in the middle lane is misrecognized as two moving objects. They are merged as one vehicle candidate, as shown in Fig. 9(b). In Fig. 9(c), the vehicles in the middle lane are misrecognized as one moving object. They are split into two vehicle candidates, as shown in Fig. 9(d).

C. Track Vehicles With Error Compensation and Update Traffic Parameters

Before applying the tracking procedure, a lane mask that was automatically built through our prior study [19] is shown in Fig. 10(a). Meaningful traffic parameters can be updated based on the detection of the lane mask. In addition, the data structure of a tracking target should be well defined for collecting the traffic parameters. These attributes, as shown in Fig. 10(b), are listed as follows.

1) *Coordinates of the left-bottom P_{LB} and the right-top P_{RT}*. The width W, height H, and gravity P_G of the tracked target can be gained by calculating P_{LB} and P_{RT} in (17).

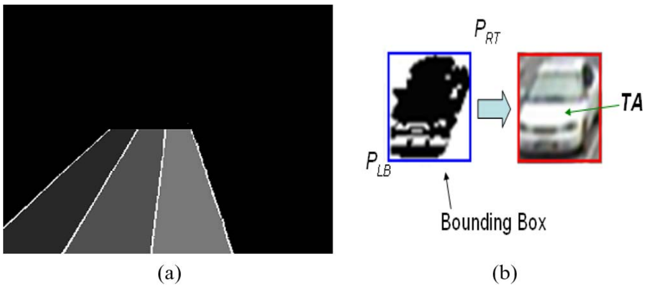

Fig. 10. Setting for tracking targets. (a) Lane masks built through [19]. (b) Attributes of tracked targets.

2) *Bounding box formed by P_{LB} and P_{RT}*. A vector of $\boldsymbol{TA(x,y)}$ is defined as the color value of (x,y) in the bounding box formed by P_{LB} and P_{RT}. In the initial state, all pixels in $\boldsymbol{TA}$ are set to $[-1,-1,-1]^T$. When a target is tracked, $\boldsymbol{TA}$ will be updated with (18). $\boldsymbol{TA}$ will be referred to in the error compensation procedure.

3) *Located lane*. The located lane can be used for calculating traffic information in each lane.

4) *Tracking and mistracking counts*. The tracking count is denoted as n, and the mistracking count is denoted as m.

5) *Current and average velocity*. The current velocity of the tracking target is denoted as $\boldsymbol{V_C} = [V_{CX}, V_{CY}]$, and the average velocity is denoted as $\boldsymbol{V_M} = [V_{MX}, V_{MY}]$, which is derived from (19), shown below

$$\begin{aligned} W &= P_{RT}(x) - P_{LB}(x) \\ H &= P_{RT}(y) - P_{LB}(y) \\ P_G &= \frac{(P_{LB}+P_{RT})}{2} \end{aligned} \tag{17}$$

$$TA_n(x,y) = \frac{n-1}{n} TA_{n-1}(x,y) + \frac{1}{n} P_n(x,y) \tag{18}$$

where n is the tracking count, $TA_n(x,y)$ is the color value of (x,y) at tracking count n, and $P_n(x,y)$ is the color value in the original frame at tracking count n. We have

$$V_M(n) = \frac{n-1}{n} V_M(n-1) + \frac{1}{n} Vc(n). \tag{19}$$

Vehicle tracking plays an important role in updating traffic parameters. The quality of traffic information is determined by the tracking methods. Fig. 11 shows the proposed tracking procedure. When vehicle candidates are detected, they are correlated with the existing tracking targets by checking the weighted determination function, denoted as DS_{max} in (20). If the detected candidate has a high correlation with the existing target, the parameters of the tracking target will be updated. If the detection candidates do not correlate with any target, they will be compared with the existing tracking acquisition targets, which have yet to be identified as tracking targets. These targets may just be noise, and hence, they need to have enough tracking information to ensure that is not the case. Once targets under tracking acquisition meet adequate appearing count, new tracking targets will be created. Otherwise, the parameters of the correlated tracking acquisition target should be updated. If vehicle candidates cannot hit any tracking or tracking acquisition targets, new tracking acquisition targets will be created.

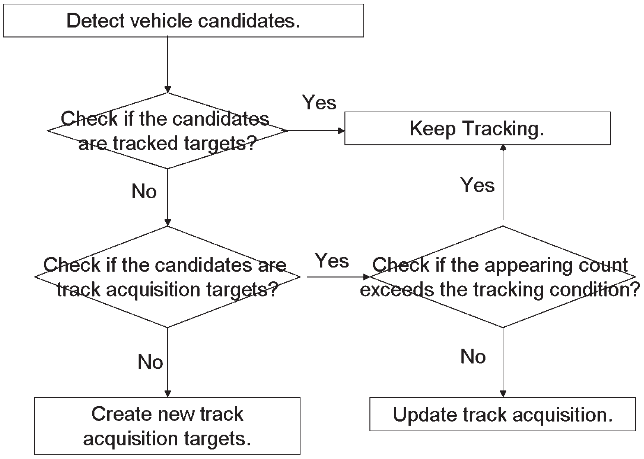
Detect vehicle candidates.
Check if the candidates are tracked targets?
Yes
Keep Tracking.
No
Yes
Check if the candidates are track acquisition targets?
Yes
Check if the appearing count exceeds the tracking condition?
No
No
Create new track acquisition targets.
Update track acquisition.

Fig. 11. Proposed tracking procedure.

TABLE III
WEIGHTING VALUES FOR TRACK ACQUISITION AND TRACK TARGETS

	α_1	α_2	α_3	TH_{DS}
Tracking Acquisition	0.33	0.33	0.33	0.7
Tracking	0.4	0.3	0.3	0.8

In (20), shown below, the weighting values of α_1, α_2, and α_3 are determined by a fuzzy decision rule. Different weighting values between tracking and tracking acquisition targets are listed in Table III. The most unlikely feature between tracking and tracking acquisition is that tracking targets have more precise gravity values. Therefore, the tracking targets have higher weighting values on ΔP_G. Finally, the target is correlated when $Hit = 1$, i.e.,

$$\begin{aligned}
M_G &= (M_{GX}, M_{GY})\\
P_G(i) &= (P_{GX}(i), P_{GY}(i))\\
\Delta P_{GX}(i) &= |P_{GX}(i) - M_{GX}|\\
\Delta P_{GY}(i) &= |P_{GY}(i) - M_{GY}|\\
\Delta P_G(i) &= \begin{cases} 1, & \Delta P_{GX}(i) \le \frac{W(i)}{2} \text{ and} \\ & \Delta P_{GY}(i) \le \frac{H(i)}{2} \\ 0, & \text{otherwise} \end{cases}\\
\Delta W(i) &= 1 - \left|\frac{W(i) - M_W}{W(i)}\right|\\
\Delta H(i) &= 1 - \left|\frac{H(i) - M_H}{H(i)}\right|\\
DS(i) &= \alpha_1 \times \Delta P_G(i) + \alpha_2 \times \Delta W(i) + \alpha_3 \times \Delta H(i)\\
DS_{\max} &= \max(DS(i))\\
Hit &= \begin{cases} 1, & DS_{\max} \ge TH_{DS} \\ 0, & \text{otherwise} \end{cases}
\end{aligned} \tag{20}$$

where M_G, M_W, and M_H are the gravity, width, and height of the detected moving object. $P_G(i)$, $W(i)$, and $H(i)$ are the gravity, width, and height of the ith tracking target. α_1, α_2, and α_3 are weighting values for the gravity, width, and height. TH_{DS} is the correlation threshold.

Some targets may be misdetected due to capturing noises or by just leaving the ROI. An error compensation flow shown in Fig. 12 is applied. First, all undetected tracking targets should be checked if they leave the ROI. When a tracking target leaves the ROI, it should be removed from the tracking list, and the traffic parameters should be updated. When a vehicle is on track, its motion should not rapidly change in a short time. If the target does not leave the ROI, the target should be redetected in the original image within a searching range based on its color information, denoted as $\boldsymbol{TA}$ in (18). The searching rule is based on (21), shown below. In (21), (x_c, y_c) is the best predicting position in the searching range

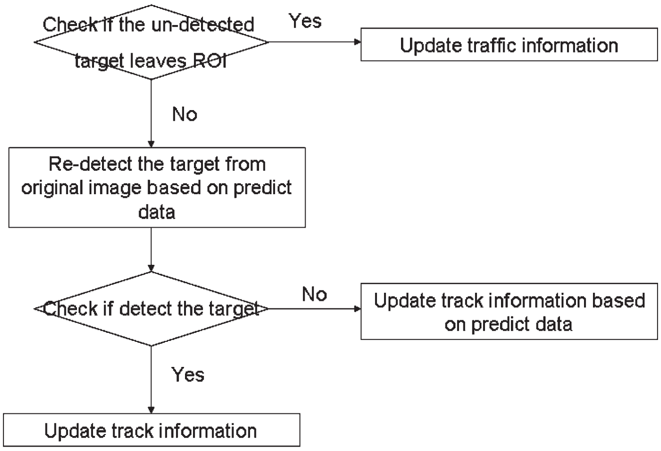
Check if the un-detected target leaves ROI
Yes
Update traffic information
No
Re-detect the target from original image based on predict data
Check if detect the target
No
Update track information based on predict data
Yes
Update track information

Fig. 12. Proposed error compensation procedure.

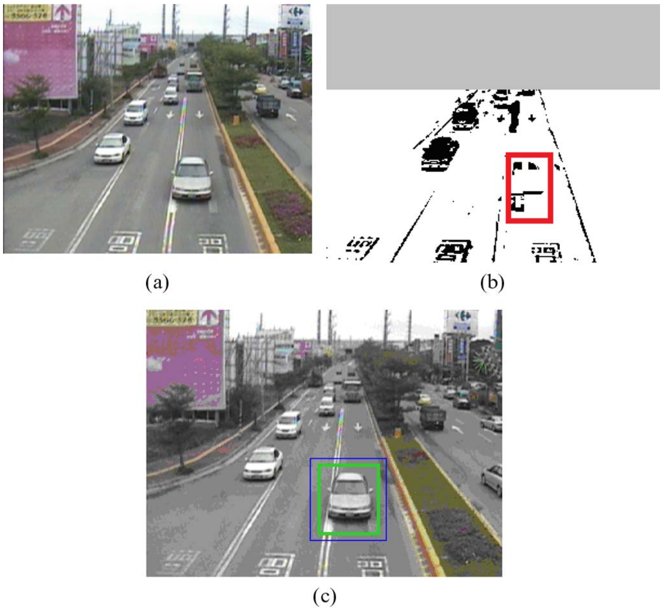
(a)
(b)
(c)

Fig. 13. Examples of the error compensation procedure. (a) Original image. (b) Vehicle is misdetected. (c) Re-searching the vehicle in the original image.

$$\begin{aligned}
S(i,j) &= \sum_{x=x_0}^{x_1} \sum_{y=y_0}^{y_1} |f(x+i, y+j) - TA(x,y)|\\
(xc, yc) &= \arg_{i,j}(\min(S(i,j)))
\end{aligned} \tag{21}$$

where i is the searching index for the horizontal with searching range $[-M, M]$, and j is the searching index for the vertical with searching range $[-N, N]$. $f(x,y)$ is the pixel value of the original image. (x_0, y_0) is the left-bottom point, and (x_1, y_1) is the right-top point of the undetected tracking target. Fig. 13

shows an example of the error compensation procedure. In Fig. 13(b), the vehicle in the rectangle is misdetected, and the predicted position does not leave the ROI; hence, the missing vehicle will be re-searched in the blue rectangle, as shown in Fig. 13(c). In Fig. 13(c), the green rectangle is the predicted car size, and the blue rectangle is the searching zone. If searching still fails, the target data will be updated with the latest predicting data. If the targets miss tracking more than three times, they will be removed from the tracking lists.

Vehicle classification plays an important role in the evaluation of traffic parameters. There are two types of vehicle classifications. The first type involves large vehicles, including buses and trucks, and the second type considers small vehicles, including sedans and vans. The determinants for small cars are based on (22), shown below. If a vehicle satisfies the conditions in (22), it will be recognized as a small car; otherwise, it will be recognized as a large car

$$\begin{cases} W < 0.8 \times W_L \\ H < 3 \times W_L \end{cases} \tag{22}$$

where W is the width of the vehicle, H is the height of the vehicle, and W_L is the width of the lane derived from [19].

V. Experimental Results

To analyze the performance of the proposed methods, we must prepare several scenarios in complex environments and in different road sections, including the highway and urban sections. These scenarios are tested on Windows XP platform with a Pentium 4 2.8-GHz central processing unit (CPU), 2-GB random access memory (RAM). The size of each image is 320 × 240, and the sampling rate of the sequence is 30 ft/s. There are four parts in the experimental results. Analyses for detecting vehicles with and without error compensation methods are addressed in Part A. Parts B and C show the accuracy ratios for velocity and vehicle classification. Finally, a comparison with other methods is presented in Part D. Testing scenarios are shown in Fig. 14. In Fig. 14(a) and (b), there are two different weather conditions: One is sunny, and the other is cloudy. Fig. 14(c) shows vehicles moving at high speed with heavy shadow effects in the highway. The rainy condition is tested in Fig. 14(d). One of the tougher test cases is presented in Fig. 14(e). Vehicles moving at night are detected in this case. Finally, test cases in traffic jams are shown in Fig. 14(g). The testing conditions for each testing scenario are listed in Table IV, where H_C is the setting height of the camera, θ_C is the view angle, V_M is the average velocity for each car during testing, and T_{AP} is the average processing time per frame. In Table IV, the average processing time per frame with a resolution of 320 × 240 in various environments is less than 13 ms, which achieves a frame rate of 76. In a real-time constraint, the processing time per frame does not exceed 50% of the CPU time. It indicates that the proposed system works well in real time.

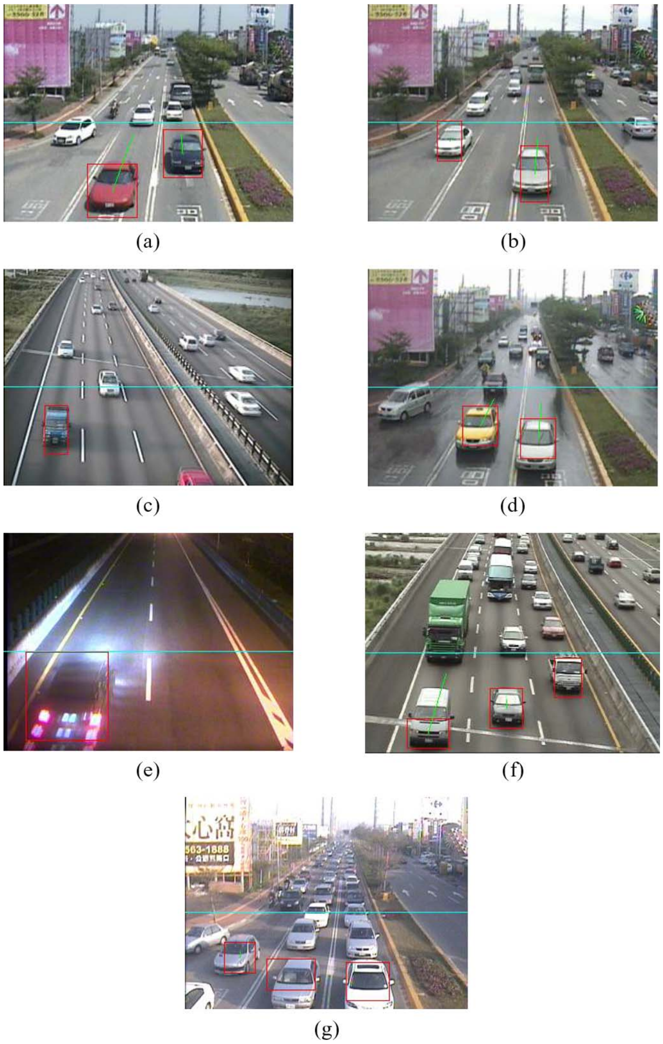
(a)
(b)
(c)
(d)
(e)
(f)
(g)

Fig. 14. Testing scenarios in various conditions. (a) Sunny day. (b) Cloudy day. (c) Shadow effects. (d) Rainy day. (e) Nighttime. (f) Heavy traffic on a highway. (g) Traffic jams in urban setting.

TABLE IV
CONDITIONS FOR EACH TESTING SCENARIO

Scenarios	H_C (m)	θ_C (°)	V_M (km/hr)	T_{AP} (ms)
(a) Sunny Day	6	10	36	11.8
(b) Cloudy Day	6	10	34	11.7
(c) Shadow Effects	12	6	89	13.4
(d) Rainy Day	6	10	33	12.3
(e) Night Time	6	10	76	11.7
(f) Heavy Traffic	12	10	45	12.7
(g) Traffic Jams	6	6	12	12.6

A. Analyses of Vehicle Detection and False Alarm

Table V shows the experimental results for detecting vehicles. The table includes the results with and without applying error compensation procedure. The detection ratios of the first two scenarios [see Table V, scenarios (a) and (b)] are similar and high. Based on the results, it can be noted that the weather effects can smoothly be overcome after applying HE. In Table V, scenario (c), we can see that the background in highway is simpler than in urban setting. The test result has a

TABLE V
EXPERIMENTAL RESULTS OF THE VEHICLE DETECTION RATIO

Scenario	Lane and Total	DC/TTC(*)	Detection Ratio (%)	DC/TTC	Detection Ratio (%)
		Without compensation		With compensation	
(a) Sunny Day	Left	144/150	96%	149/ 150	99.33%
	Middle	187/191	97.9%	189/191	98.95%
	Right	148/150	98.67%	150/150	100%
	Total	479/491	97.55%	488/491	99.38%
(b) Cloudy Day	Left	188/194	96.9%	192/194	98.96%
	Middle	192/197	97.46%	196/197	99.49%
	Right	207/212	97.64%	210/212	99.06%
	Total	587/603	97.35%	598/603	99.17%
(c) Shadow Effects	Left	206/210	98.1%	209/ 210	99.52%
	Middle	248/251	98.8%	251/251	100%
	Right	245/250	98%	247/250	98.8%
	Total	699/711	98.3%	699/711	99.4%
(d) Rainy Day	Left	56/65	86.1%	60/65	92.3%
	Middle	61/70	87.1%	66/70	94.3%
	Right	42/50	84%	46/50	92%
	Total	159/185	85.9%	172/185	93%
(e) Night time	Left	98/122	80.32%	115/122	94.26%
	Right	150/180	83.3%	168/180	93.3%
	Total	248/302	82.1%	283/302	93.7%
(f) Heavy Traffic	Left	165/174	94.8%	168/174	96.5%
	Middle	186/192	96.9%	188/192	97.9%
	Right	185/190	97.36%	186/190	97.9%
	Total	536/556	96.4%	542/556	97.5%
(g) Traffic Jams	Left	55/58	94.8%	57/58	98.2%
	Middle	51/54	94.4	53/54	98.1
	Right	65/67	97%	66/67	98.5%
	Total	171/179	95.5%	176/179	98.3%

(*) DC is the Detection Count and TTC is the Total Target Count.

TABLE VI
EXPERIMENTAL RESULTS OF THE FAR

Scenario	Lane and Total	FAC (*)	FAR(**) (%)	FAC	FAR (%)
		Without compensation		With compensation	
(a) Sunny Day	Left	2	1.33%	2	1.33%
	Middle	3	1.57%	3	1.57%
	Right	2	1.33%	2	1.33%
	Total	7	1.43%	7	1.43%
(b) Cloudy Day	Left	3	1.54%	3	1.54%
	Middle	4	2.03%	4	2.03%
	Right	3	1.41%	3	1.41%
	Total	10	1.66%	10	1.66%
(c) Shadow Effects	Left	10	4.76%	10	4.76%
	Middle	15	5.98%	15	5.98%
	Right	3	1.20%	3	1.20%
	Total	28	3.94%	28	3.94%
(d) Rainy Day	Left	1	1.53%	1	1.53%
	Middle	2	2.86%	2	2.86%
	Right	1	2.00%	1	2.00%
	Total	4	2.16%	4	2.16%
(e) Night time	Left	10	8.20%	10	8.20%
	Right	13	7.22%	13	7.22%
	Total	23	7.62%	23	7.62%
(f) Heavy Traffic	Left	9	5.18%	9	5.18%
	Middle	10	5.21%	10	5.21%
	Right	3	1.58%	3	1.58%
	Total	22	3.96%	22	3.96%
(g) Traffic Jams	Left	2	3.44%	2	3.44%
	Middle	1	1.85%	1	1.85%
	Right	2	2.99%	2	2.99%
	Total	5	2.79%	5	2.79%

(*) FAC is the False Alarm Count.
(**) FAR is the False Alarm Ratio, FAR=FAC/TTC.

higher detection ratio in this case than in the urban. In Table V, scenario (d), it can be noted that the detection ratio in the rainy condition can achieve around 85% without error compensation. After applying the error compensation procedure, the detection ratio can rise to 93%. In addition, the detection ratio at night can only reach around 82%. Again, the error compensation can reach the ratio to 93.7%. These two test scenarios are found to have lower detection ratios without the application of the error compensation procedure. Finally, the results of the two traffic jam scenarios are presented, and they show that the proposed system can do well in traffic jam conditions, particularly on highways.

The experimental results of the false alarm (which are also known as false-positive detection) for various scenarios are listed in Table VI. A detected vehicle should satisfy the tracking procedure shown in Fig. 11; therefore, the false alarms seldom appear in the proposed system. However, few false alarms, including reels or large trunks, may be misrecognized as two vehicles in some cases. Fig. 15 is an example of a false alarm induced by a large trunk. Based on the experimental results in Table VI, the proposed method has low false-alarm counts (FAC) and false-alarm ratios (FAR) in most scenarios. At night, [see Table V, scenario (e)], vehicle-light effects will induce some FARs. In addition, more false alarms, including reels and large trunks, may appear in the highway; therefore, the FAR will slightly be impacted. In the highway, reels and large trunks are not allowed to move in the right lane, therefore, the most false detections appear in the left and middle lanes.

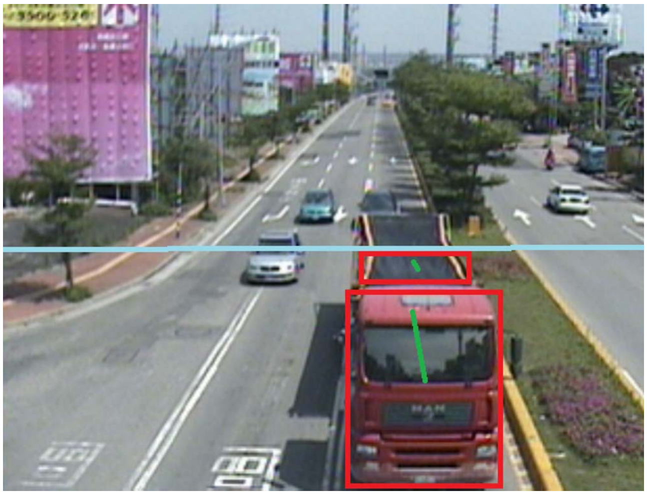

Fig. 15. Large trunk induces false-alarm detection.

B. Accuracy Ratio of Vehicle Velocity

Experimental results for the detection of vehicle velocity are shown in Table VII. The initial velocity of the tracking target is calculated from the change in gravity. Then, the average velocity, which is denoted as V_M, is updated with (19). The reference velocity is detected by the velocity-detection radar. The tolerance is set to ±5 km/h. If the difference between the velocity detected by the proposed system and the velocity detected by radar is lower than the tolerance, the velocity can be thought to be correct. In Table VII, two types of accuracy

TABLE VII
ACCURACY RATIO FOR VEHICLE VELOCITY

Scenario	Lane and Total	DCV/TTC (*)	Accuracy Ratio (%)	DCV/DC	Accuracy Ratio (%)
(a) Sunny Day	Left	143/150	95.3%	143/149	96%
	Middle	184/191	96.3%	184/189	97.4%
	Right	144/150	96.0%	144/150	96%
	Total	471/491	95.9%	471/488	96.5%
(b) Cloudy Day	Left	185/194	95.3%	185/192	96.4%
	Middle	188/197	95.4%	188/196	96%
	Right	195/212	92%	195/210	92.9%
	Total	568/603	94.2%	568/598	95%
(c) Shadow Effects	Left	198/210	94.3%	198/209	94.7%
	Middle	244/251	97.2%	244/251	97.2%
	Right	239/250	95.6%	239/247	96.8%
	Total	681/711	95.8%	681/699	97.4%
(d) Rainy Day	Left	57/65	87.7%	57/60	95%
	Middle	58/70	82.9%	58/66	87.9%
	Right	40/50	80.0%	40/46	87%
	Total	155/185	83.8%	155/172	90.1%
(e) Night time	Left	98/122	80.3%	98/115	85.2%
	Right	139/180	77.2%	139/168	82.7%
	Total	237/302	78.5%	237/283	83.7%
(f) Heavy Traffic	Left	158/174	90.8%	158/168	94%
	Middle	172/192	89.6%	172/188	91.5%
	Right	180/190	94.7%	180/186	96.8%
	Total	510/556	91.7%	510/542	94.4%
(g) Traffic Jams	Left	48/58	82.8%	48/57	84.2%
	Middle	47/54	87%	47/53	88.7%
	Right	55/67	82.1%	55/66	83.3%
	Total	150/179	83.8%	150/176	85.2%

(*) DCV is the correct detection count for velocity.

TABLE VIII
ACCURACY RATIO FOR VEHICLE CLASSIFICATION

Scenario	CR/TTC (*)	Accuracy Ratio (%)	CR/DC	Accuracy Ratio (%)
(a) Sunny Day	451/491	91.9%	451/488	92.4%
(b) Cloudy Day	551/603	91.4%	551/598	92.1%
(c) Shadow Effects	640/711	90.0%	640/699	91.6%
(d) Rainy Day	150/185	81.1%	150/172	87.2%
(e) Night Time	211/302	69.9%	211/283	74.6%
(f) Heavy Traffic	492/556	88.5%	492/542	90.8%
(g) Traffic Jams	170/179	95.0%	170/176	96.6%

(*) CR is the Accuracy Ratio for vehicle classification.

TABLE IX
COMPARISONS WITH OTHER APPROACHES

Scenarios	Type	Cucchiara et al. [14]	Gupte et al. [18]	Yang [1]	Proposed System
(a) Sunny Day	DC/TTC	97.8%	96.8%	98.5%	99.38%
	CR/DC	N/A	92.5%	N/A	92.4%
	DCV/DC	N/A	N/A	N/A	96.5%
(b) Cloudy Day	DC/TTC	97.6%	98.87%	99.2%	99.17%
	CR/DC	N/A	91.5%	N/A	92.1%
	DCV/DC	N/A	N/A	N/A	95%
(c) Shadow Effects	DC/TTC	95.8%	94.8%	96.5%	99.4%
	CR/DC	N/A	90.2%	N/A	91.6%
	DCV/DC	N/A	N/A	N/A	97.4%
(d) Rainy Day	DC/TTC	91.5%	92.5%	94.2%	93%
	CR/DC	N/A	88.6%	N/A	87.2%
	DCV/DC	N/A	N/A	N/A	90.1%
(e) Night Time	DC/TTC	78.6%	70.5%	89.2%	93.7%
	CR/DC	N/A	70.5%	N/A	74.6%
	DCV/DC	N/A	N/A	N/A	83.7%
(f) Heavy Traffic	DC/TTC	96.2%	90.2%	96.8%	97.5%
	CR/DC	N/A	85.5%	N/A	90.8%
	DCV/DC	N/A	N/A	N/A	94.4%
(g) Traffic Jams	DC/TTC	96.8%	92.5%	97.2%	98.3%
	CR/DC	N/A	90.6%	N/A	96.6%
	DCV/DC	N/A	N/A	N/A	85.2%

ratios are calculated. One type is based on the total target count (TTC), which will be affected by the detection ratios shown in Table V. The other type is based on the detected count (DC), and the detection ratios are higher than the former type. Based on the results listed in the table, the detection ratios of velocity are high in most cases. In the night case of Table VII, scenario (e), the detection ratio is lower than other cases, because the light effects greatly influence size detection and gravity position. In the traffic jam case of Table VII, scenario (g), vehicles move slowly and suddenly stop for the traffic light. Therefore, the velocities in this case heavily vibrate, and the accuracy ratio is lower than other cases.

C. Accuracy Ratio of Vehicle Classification

Table VIII shows the accuracy ratios for vehicle classification. There are two types of vehicles to classify in (22). The first type is large vehicles, including buses and trucks, and the other type is small vehicles, including sedans and vans. The major differences between small and large vehicles are the width and the height. Based on the experimental result in Table VIII, the classification results are good, except for the cases in rain and at night.

D. Comparison With Other Approaches

A comparison with other approaches is listed in Table IX. As shown, the accuracy ratios for detecting vehicles in [1] are similar to the proposed system. However, it does not detect the vehicle velocity and vehicle classifications. The detection ratios in [14] are lower than the proposed system. In addition, it evaluates fewer traffic parameters. The detection ratios in [18] are also lower than the proposed approach. It also calculates fewer traffic parameters.

VI. CONCLUSION

In this paper, an adaptive vehicle detection approach for complex environments has been presented. This paper has also proposed methods for solving vehicle tracking in traffic jams and complex weather conditions, such as sunny, rain, sunrise, sunset, cloudy, or snowy days. HE is used to remove the effects of weather and light impact. The method is applied to improve the tracking accuracy ratio and simplify the system parameter settings. GDVM is used to dynamically segment moving objects. Finally, tracking and predict compensation are applied to refine the target tracking quality. Based on the experimental results, the data indicate that the tracking accuracy ratio of the proposed system is quite good in traffic jams and complex weather conditions, particularly when applying the error compensation procedure. In the comparisons with other approaches, the proposed method not only has higher detection ratios but gathers more useful traffic parameters as well. In addition, the proposed system can easily be set up without being given any environment information in advance. In this paper, many useful traffic parameters are built, and they can be used to control the traffic. Furthermore, this information can be combined with a personal digital assistant (PDA) or mobile phone system to provide traffic conditions for vehicle drivers. In future works, we still need to improve the accuracy ratio when it is raining and at night. In addition, the detection of motorcycles is necessary to make the system practical for commercial usage.

REFERENCES

[1] 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.

[2] L.-W. Tsai, J.-W. Hsieh, and K.-C. Fan, "Vehicle detection using normalized color and edge map," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 850–864, Mar. 2007.

[3] W. Zhang, Q. M. J. Wu, and X. Yang, "Multilevel framework to detect and handle vehicle occlusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 161–174, Mar. 2008.

[4] 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.

[5] 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.

[6] H.-Y. Cheng, B.-S. Jeng, P.-T. Tseng, and K.-C. Fan, "Lane detection with moving vehicles in the traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 571–582, Dec. 2006.

[7] M. Vargas, J. M. Milla, S. L. Toral, and F. Barrero, "An enhanced background estimation algorithm for vehicle detection in urban traffic scenes," *IEEE Trans. Veh. Technol.*, vol. 59, no. 8, pp. 3694–3709, Oct. 2010.

[8] 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.

[9] D. Koller, K. Daniilidis, and H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, Jun. 1993.

[10] J. C. Tai, S. T. Tseng, C. P. Lin, and K. T. Song, "Real-time image tracking for automatic traffic monitoring and enforcement applications," *Image Vis. Comput.*, vol. 22, no. 6, pp. 485–501, Jun. 2004.

[11] D. Koller, J. Weber, T. Huang, J. Malik, J. Ogasawara, G. Rao, and S. Russell, "Towards robust automatic traffic scene analysis in real-time," in *Proc. 12th Int. Conf. Comput. Vis. Image Process.*, 1994, vol. 1, pp. 126–131.

[12] B. Coifman, D. Beyber, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transp. Res. Part C*, vol. 6, no. 4, pp. 271–288, 1998.

[13] Y. K. Wang and S. H. Chen, "A robust vehicle detection approach," in *Proc. IEEE Conf. Adv. Video Signal Based Surveillance*, 2005, pp. 117–122.

[14] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 119–130, Jun. 2000.

[15] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measure traffic parameters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, San Juan, PR, Jun. 1997, pp. 496–501.

[16] W. L. Hsu, H. Y. Liao, B. S. Jeng, and K. C. Fan, "Real-time traffic parameter extraction using entropy," *Proc. Inst. Elect. Eng.—Vis. Image Signal Process.*, vol. 151, no. 3, pp. 194–202, Jun. 2004.

[17] G. D. Sullivan, K. D. Baker, A. D. Worrall, C. I. Attwood, and P. M. Remagnino, "Model-based vehicle detection and classification using orthographic approximation," *Image Vis. Comput.*, vol. 15, no. 8, pp. 649–654, Aug. 1997.

[18] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 37–47, Mar. 2002.

[19] B. F. Wu, S. P. Lin, and Y. H. Chen, "A real-time multiple-vehicle detection and tracking system with prior occlusion detection and resolution and prior queue detection and resolution," in *Proc. 18th ICPR*, Aug. 2006, vol. 1, pp. 828–831.

[20] B. Bartin and K. Ozbay, "Determining the optimal configuration of highway routes for real-time traffic information: A case study," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 225–231, Mar. 2010.

[21] H. Xie, L. Kulik, and E. Tanin, "Privacy-aware traffic monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 61–70, Mar. 2010.

[22] T.-H. Chang and H.-J. Huang, "Dynamic data regulation for fixed vehicle detectors," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 113–121, Mar. 2010.

[23] D. I. B. Randeniya, S. Sarkar, and M. Gunaratne, "Vision-IMU integration using a slow-frame-rate monocular vision system in an actual roadway setting," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 256–266, Jun. 2010.

[24] N. K. Kanhere and S. T. Birchfield, "A taxonomy and analysis of camera calibration methods for traffic monitoring applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 441–452, Jun. 2010.

[25] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[26] J. H. Lim, Y. Liu, and O. Tsimhoni, "Investigation of driver performance with night-vision and pedestrian-detection systems—Part 2: Queuing network human performance modeling," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 765–772, Dec. 2010.

[27] 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.

[28] 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.

Bing-Fei Wu (S'89–M'92–SM'02–F'12) was born in Taipei, Taiwan, in 1959. He received the B.S. and M.S. degrees in control engineering from the National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1992.

Since 1992, he has been with the Department of Electrical Engineering, NCTU, where he became a Professor in 1998 and was promoted to Distinguished Professor in 2010. Since 2011, he has served as the Director of the Institute of Electrical and Control Engineering, NCTU. He has been involved in research on intelligent transportation systems for many years and leads a research team to develop the first Taiwan smart car, i.e., TAIWAN *i*TS-1, with autonomous driving and active safety system. His current research interests include image recognition, vehicle driving safety, intelligent control, intelligent transportation systems, multimedia signal analysis, embedded systems, and chip design.

Dr. Wu is a Fellow of the Institution of Engineering and Technology. He founded and served as the Chair of the IEEE Systems, Man, and Cybernetics (SMC) Society Taipei Chapter in Taiwan in 2003. He was also elected as the Chair of the Technical Committee on Intelligent Transportation Systems of IEEE SMC Society in 2011. He has been the Director of The Research Group of Control Technology of Consumer Electronics for the Automatic Control Section of the National Science Council, Taiwan, from 1999 to 2000. He received of the Golden Acer Dragon Thesis Award sponsored by the Acer Foundation in 1998 and 2003, the Distinguished Engineering Professor Award from the Chinese Institute of Engineers in 2002, the Outstanding Information Technology Elite Award from the Taiwan Government in 2003, the Outstanding Research Award from NCTU in 2004, the First Prize Award of the TI China-Taiwan DSP Design Contest in 2006, and the Outstanding Automatic Control Engineering Award from the Chinese Automatic Control Society and the Fuji/Xerox Academic Research Award in 2007.

Jhy-Hong Juang was born in I-Lan, Taiwan, in 1974. He received the B.S. degree in control engineering and the M.S. degree from the National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1997 and 1999, respectively. He is currently working toward the Ph.D. degree with the Institute of Electrical and Control Engineering, NCTU.

His research interests include image processing, software engineering, and pattern recognition.