Intelligent Highway Traffic Surveillance With Self-Diagnosis Abilities

Hsu-Yung Cheng, *Member, IEEE*, and Shih-Han Hsu

***Abstract*—In this paper, we propose a self-diagnosing intelligent highway surveillance system and design effective solutions for both daytime and nighttime traffic surveillance. For daytime surveillance, vehicles are detected via background modeling. For nighttime videos, headlights of vehicles need to be located and paired for vehicle detection. An algorithm based on likelihood computation is developed to pair the headlights of vehicles at night. Moreover, to balance between the robustness and abundance of acquired information, the proposed system adapts different strategies under different traffic conditions. Performing tracking would be preferred when traffic is smooth. However, under congestion conditions, it is better to obtain traffic parameters by estimation. We utilize a time-varying adaptive system state transition matrix in Kalman filter for better prediction in a traffic surveillance scene when performing tracking. We also propose a mechanism for estimating the traffic flow parameter via regression analysis. The experimental results have shown that the self-diagnosis ability and the modules designed for the system make the proposed system robust and reliable.**

***Index Terms*—Headlight pairing, intelligent surveillance, regression analysis, tracking, traffic parameter.**

I. Introduction

TRAFFIC surveillance plays an important role in intelligent transportation systems, and vision-based techniques have become current trends [1]–[3]. From recent research works and commercial systems, it has been shown that a wide range of information can be obtained from cameras, including vehicle classification, incidents, rule violation, turning behaviors, etc. [4], [5]. Using cameras as sensors brings in a large variety of advantages but faces many challenges as well. The challenges include inconsistent surveillance scenes due to changing lighting conditions and occlusions among vehicles due to different camera angles or traffic congestion. To overcome these challenges, specialized procedures can be designed to deal with specific lighting or traffic conditions. As a result, a vision-based intelligent system should be able to perform self-diagnosis to observe the current condition of the surveillance video and decide the best procedure to follow. In this paper, we propose a self-diagnosing intelligent highway surveillance system and design effective solutions for both daytime and nighttime traffic surveillance. Moreover, to balance between the robustness and abundance of acquired information, the proposed system adopts tracking and regression-based traffic parameter estimation under different traffic conditions. Such a mechanism enhances the reliability of the system.

As we know, daytime and nighttime videos require very different handling processes. Therefore, vision-based intelligent surveillance systems should be able to determine if daytime modules or nighttime modules should be executed. For nighttime surveillance videos, it is difficult to segment the entire vehicles from the background. Foreground objects that can be detected at night are usually headlights, auxiliary lights, or reflections of lights of the vehicles. Headlights are important features for initializing vehicles for tracking at night. Therefore, locating and pairing headlights are important for nighttime surveillance. In [5], separated low-level modules were used for daytime and nighttime videos. Nighttime vehicles are detected by performing headlight pairing using template matching on the image frames with a headlight template, which was scaled with respect to the image regions. More headlight pairing algorithms can be found in more recent works [6]–[9]. Chen *et al.* [6] proposed a nighttime vehicle detection algorithm for driver assistance systems. They applied spatial clustering to look for horizontal aligned vehicle lights among all detected lighting objects. Rule-based vehicle identification is used in [6]. Rule-based methods are feasible because the number of target vehicles is small in driver assistance applications. In [7], O'Malley *et al.* proposed a system to detect and track rear-lamp pairs. However, it is also designed mainly for driver-assistance systems and not surveillance systems. In surveillance applications, such methods would not be appropriate. Robert [8] presented a framework to detect and track multiple vehicles for nighttime traffic surveillance. They invoked an extra stage to identify the vehicles using a decision tree composed of feature- and appearance-based classifiers to avoid false alarms. Two weak classifiers and one support vector machine or artificial neural network are required to be trained for vehicle identification in [8]. In addition, it is necessary to collect vehicle samples at night to form the eigenvehicles. For different camera angles and surveillance scenes, the vehicle samples would need to be manually cropped and collected to retrain the classifier. Wang *et al.* [9] also used a training-based method to form a two-layer nighttime vehicle detector using AdaBoost cascade classifiers. The main disadvantage of [9] is that the method needs

Manuscript received July 15, 2010; revised April 19, 2011; accepted June 4, 2011. Date of publication September 22, 2011; date of current version December 5, 2011. This work was supported in part by the National Science Council of Taiwan under Project 99-2628-E-008-098. The Associate Editor for this paper was Z. Li.

The authors are with the Department of Computer Science and Information Engineering, National Central University, Chung-li 320, Taiwan (e-mail: chengsy@csie.ncu.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.2011.2160171

many positive and negative samples. The classifiers would also need to be retrained for different camera angles. In this paper, to more effectively detect vehicles at nighttime, an algorithm based on likelihood computation is developed to pair the headlights. Through the pairing procedure, the headlights can also be distinguished from other auxiliary lights and reflections of lights of the vehicles. The proposed headlight pairing method is more flexible and effective than the method using template matching in [5]. In addition, there is no need to manually collect vehicle samples at night, as required in [8] or [9]. For different surveillance scenes, instead of recollecting the vehicle image samples and retraining the classifiers, the proposed method only requires remeasuring the pixel distance between headlights at certain positions of the scene, which would be much easier.

The main functions of the intelligent traffic surveillance systems are to obtain traffic parameters and to detect events of interest. If tracking algorithms could properly work, performing tracking should be preferred in intelligent surveillance systems. The reasons are twofold: First, from satisfactory tracking results, accurate traffic parameters can be extracted [10]. In addition, tracking results provide the detailed behavior and trajectory of each individual vehicle, which could contribute more useful information to subsequent event detection modules, such as the systems developed in [11]–[13]. Moreover, with the help of a vehicle tracking system, another camera can zoom in and capture the license plate of the tracked vehicle in the case of rule violations such as an illegal lane change, driving on a freeway shoulder, or when heavy vehicles driving on inner lanes are detected. Researchers have devoted themselves in developing robust tracking algorithms that can deal with occlusions and segmentation errors [14]–[18]. In this paper, we incorporate an efficient tracking mechanism that utilizes Kalman filtering, adaptive particle sampling, and enhanced probabilistic data association to track the vehicles [16], [18]. The main improvement on tracking in this work is that the system state transition model of the Kalman filter (KF) is specialized to adapt to common settings of highway traffic surveillance cameras. The adaptive system transition matrix results in better prediction for tracking. For vehicles undergoing temporary interobject occlusion in the surveillance scene, adaptive particle sampling and enhanced probabilistic data association are able to help resolve the occlusion conditions. However, it is unrealistic to segment and track each individual vehicle under all circumstances. Under congestion, initial occlusion would make segmentation and initialization of target objects very difficult. In addition to unstable segmentation and initialization of objects, vehicles that are seriously occluded all the way also make performing tracking impractical. Therefore, when the traffic is congested, it is better to obtain the traffic flow parameter by estimation rather than by tracking.

Traffic parameter extraction methods that are not based on tracking mechanisms can be found in the literature [19]–[24]. Fathy and Siyal [19], [20] proposed window-based techniques to extract traffic queue parameters. Viarani [21] also used preset windows called virtual inductive loops (VILs) to detect the differences caused by the foreground vehicles to extract traffic parameters. Hsu *et al.* [22] improved the window-based methods and determined traffic parameters based on entropy measurements in detection zones. In [23], the authors trained a neural network classifier to judge the length of a traffic queue. Lee and Park [24] classified local detection regions into roads, vehicles, shadows, and reflections. Afterward, they generated spatiotemporal images, and traffic parameter extraction was achieved by analyzing the spatiotemporal images. Headlight detection for nighttime operation was also employed in [24]. By examining previous works, we can observe that window-based methods are fast, but when the congestion condition becomes severe, these methods would tend to give erroneous results. To reduce the errors under severe congestion conditions, we design a new estimation mechanism via regression to estimate the number of vehicles passing a specific region within a fixed time duration in this work. The proposed regression-based method keeps the advantage of being efficient. In addition, one of the main advantages of the proposed method is that the proposed method would not be seriously affected by segmentation errors of the foreground objects. Note that the proposed headlight pairing, adaptive KF system transition model, and regression-based traffic flow estimation mechanism are designed mainly for highway traffic and not for intersections. In addition, the proposed system is tested only by surveillance videos under highway settings. The rest of this paper is organized as follows: In Section II, we describe the modules of the proposed system framework. In Section III, we explain the proposed headlight-pairing algorithm for nighttime surveillance. The vehicle tracking mechanism is elaborated in Section IV. Estimation of traffic flow parameter via regression is elaborated in Section V. We demonstrate the experimental results in Section VI. Finally, conclusions are made in Section VII.

II. Proposed System Framework

The proposed system framework is shown in Fig. 1. The first self-diagnosis step is to determine the lighting condition. Currently, two basic conditions, i.e., daytime and nighttime, are considered in our work. However, more sophisticated subdivisions, such as rainy or foggy conditions, can be defined, and the corresponding handling procedures can be designed accordingly. For daytime surveillance, foreground vehicles are detected by background modeling with shadow removal [25]. For nighttime surveillance, vehicle detection is achieved by locating and pairing headlights. For both daytime and nighttime surveillance, lane detection is performed to analyze the traffic parameters and events of each lane [26]. In addition, the features in each lane are extracted to support the second diagnosis step. The second diagnosis step is to determine whether tracking should be performed. It is very important to know if the system should trust the results of tracking. If tracking is favorable, the system tracks each individual vehicle and obtains traffic parameters from the tracking results. The tracking mechanism for the system is explained in Section IV. If tracking is inappropriate, the system performs traffic parameter extraction via regression, which is elaborated in Section V. In the succeeding sections, we describe the feature extraction and self-diagnosis modules in the system.

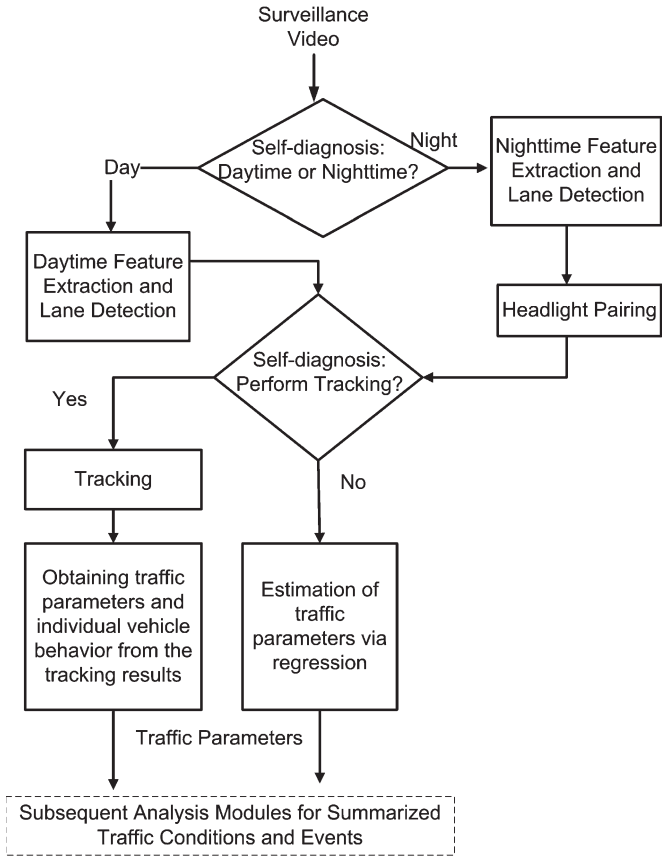

Fig. 1. Proposed system framework.

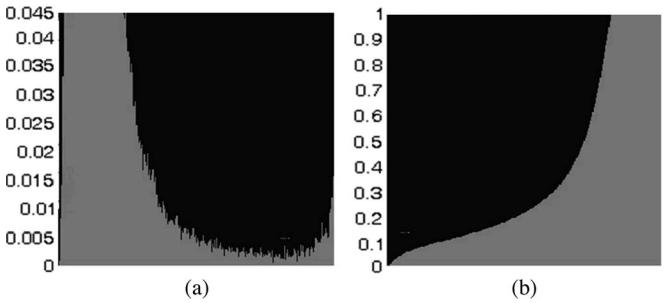

Fig. 2. Histogram analysis for dynamic threshold determination. (a) Original histogram. (b) Backward cumulative histogram.

A. Feature Extraction

The features for daytime surveillance videos are the corner and edge densities in the region of interest of each lane. The features for nighttime surveillance videos are the corner density, edge density, and the number of lighting objects in the region of interest of each lane. We apply Harris corner detector [27] for corner detection. Lighting objects are detected via thresholding. A fixed static threshold is not suitable for lighting object detection because the desired threshold varies under different lighting conditions at night. In some sections of the roads, the lighting conditions are better than others. In addition, the entire surveillance scene becomes brighter under congested conditions, compared with the conditions when vehicles are sparse on the road. Therefore, the thresholds need to be dynamically determined according to the intensity histogram of each image frame. Fig. 2(a) shows the intensity histogram of a nighttime image scene. We can observe that most intensities concentrate on lower gray levels, which correspond to the dark background at night. In addition, the lighting objects correspond to the higher intensities to the right of Fig. 2(a). The desired threshold should be the point before the values in the histogram bins surge again. However, the intensity histograms are usually rugged and irregular. Therefore, we accumulate the histogram bins $Hist(i)$ from higher intensities to lower intensities to obtain a smooth backward cumulative histogram $Hist_{\mathrm{BC}}(n)$

$$Hist_{\mathrm{BC}}(n) = \sum_{i=255}^{255-n} Hist(i). \tag{1}$$

Fig. 2(a) shows the original histogram of a frame, and Fig. 2(b) shows the backward cumulative histogram $Hist_{\mathrm{BC}}(n)$. We take the first derivative of the backward cumulative histogram, and the desired threshold $\hat{T}h$ can be found by

$$\hat{T}h = 255 - \arg\min_{\tau} \frac{\partial Hist_{\mathrm{BC}}(\tau)}{\partial \tau}. \tag{2}$$

After thresholding, the binary mask is refined using morphological operations to remove noise and to make the detected blobs more complete.

B. Self-Diagnosis

Self-diagnosis of the lighting condition is performed by examining the average intensity and the red, green, and blue (RGB) color components of the surveillance scene. We could observe that the average intensity tends to be lower and the difference between each RGB color component would be small at night. If

$$\frac{1}{N \cdot \Delta t} \sum_{t-\Delta t}^{t} \sum_{u,v} I_t(u,v) \geq Th_1 \tag{3}$$

$$\frac{1}{N \cdot \Delta t} \sum_{t-\Delta t}^{t} \sum_{u,v} \Delta_t^{\mathrm{RG}}(u,v) + \Delta_t^{\mathrm{RB}}(u,v) + \Delta_t^{\mathrm{GB}}(u,v) \geq Th_2 \tag{4}$$

are both satisfied, then daytime procedures are applied. Otherwise, nighttime time procedures are invoked. In (3), $I_t(u,v)$ denotes the intensity of pixel (u,v) at frame t. In (4), $\Delta_t^{\mathrm{RG}}(u,v)$ denotes the absolute difference between the red and green color components of pixel (u,v) at frame t; $\Delta_t^{\mathrm{RB}}(u,v)$ and $\Delta_t^{\mathrm{GB}}(u,v)$ are similarly defined. To avoid sudden change of light, we average these terms over the duration of Δt frames. Note that N denotes the number of pixels in the image frame, and Th_1 and Th_2 are two thresholds.

We use a trained two-class Bayesian classifier to conduct self-diagnosis to determine the adequacy of performing the tracking algorithm under current traffic conditions for each lane. The first-class ω_1 corresponds to performing tracking, and the second-class ω_2 corresponds to performing regression. The feature vector $\mathbf{x}$ is extracted, as described in the previous section. More specifically, $\mathbf{x}$ is a 2-D vector $(c_d e_d)^T$ for daytime

videos. In addition, $\mathbf{x}$ is a 3-D vector $(c_d e_d n_l)^T$ for nighttime videos, where c_d, e_d, and n_l denote the corner density, edge density, and the number of lighting objects in the region of interest of each lane, respectively, i.e.,

$$P(\omega_i \mid \mathbf{x}) = \frac{P(\omega_i)P(\mathbf{x} \mid \omega_i)}{P(\mathbf{x})}. \tag{5}$$

The class conditional probability density $P(\mathbf{x}|\omega_i)$ is modeled by a Gaussian function. The parameters of the Gaussian function for each class is obtained by maximum-likelihood estimation [28]. The prior probabilities of the Bayesian classifier $P(\omega_i)$ can be determined by a lookup table indexed by location and time. Here, we set simple prior probabilities by counting the number of frames of congested conditions and uncongested conditions in the training videos. More specifically, the prior probability of being under a congested condition for a certain lane L is equal to the number of frames of congested conditions for lane L divided by the total frame number.

III. Headlight Pairing

The detected blobs described in Section II-A serve as the input of the pairing algorithm. The pairing algorithm is based on likelihood computation. The likelihood that the ith blob and the jth blob belong to the same headlight pair is defined as follows:

$$P_{\text{pair}}(i,j) = P_{\text{distance}}(i,j) \cdot P_{\text{angle}}(i,j) \tag{6}$$

where $P_{\text{distance}}(i,j)$ is the likelihood computed based on the distance between the ith blob and the jth blob $Dist(i,j)$. In addition, $P_{\text{angle}}(i,j)$ is the likelihood computed based on $Angle(i,j)$, which is the angle between the horizontal line and the line segment connecting the two blobs. We define $P_{\text{distance}}(i,j)$ as follows:

$$P_{\text{distance}}(i,j) = \begin{cases} 0 & if\ Dist(i,j) > Thr_{\text{Dist}} \\ \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{(Dist(i,j)-\mu_{\text{Dist}})^2}{2\sigma_1^2}} & if\ Dist(i,j) \le Thr_{\text{Dist}}. \end{cases} \tag{7}$$

In (7), $P_{\text{distance}}(i,j)$ is equal to zero if the distance between the two blobs $Dist(i,j)$ is larger than a threshold Thr_{Dist}. For blob pairs whose distance is smaller than the threshold, the likelihood $P_{\text{distance}}(i,j)$ is defined by a Gaussian function, which has a higher value when $Dist(i,j)$ is near the mean distance μ_{Dist}. The mean distance μ_{Dist} between headlight pairs can be obtained by observing the actual headlight pairs in the surveillance scenes, which will be described later in this section. Let $u^{(i)}$ and $v^{(i)}$ denote the coordinate of the centroid of the ith blob. In addition, $u^{(j)}$ and $v^{(j)}$ denote the coordinate of the centroid of the jth blob. We define $Angle(i,j)$ as

$$Angle(i,j) = \tan^{-1}\frac{v^{(j)} - v^{(i)}}{u^{(j)} - u^{(i)}}. \tag{8}$$

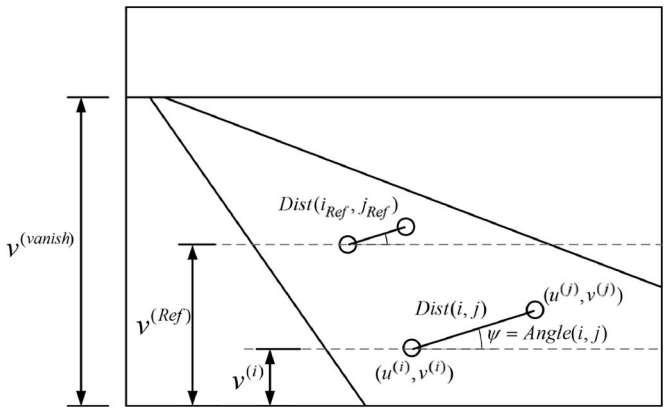

Fig. 3. Distance between headlights.

Similar to (7), we could define $P_{\text{angle}}(i,j)$ using another Gaussian as

$$P_{\text{angle}}(i,j) = \begin{cases} 0 & Angle(i,j) > Thr_{\text{Angle}} \\ \frac{1}{\sigma_2\sqrt{2\pi}} e^{-\frac{(Angle(i,j)-\mu_{\text{Angle}})^2}{2\sigma_2^2}} & Angle(i,j) \le Thr_{\text{Angle}} \end{cases} \tag{9}$$

with μ_{Angle} similarly defined. In (7) and (9), σ_1 and σ_2 are the standard deviations of the Gaussian functions.

The angle between the horizontal line and the line connecting the two headlights stays nearly the same in highway surveillance scenes. In addition, the regions with sufficient resolutions that we are interested in do not cover the distant view of the curved lane in the scene. Therefore, it is reasonable to assume that $Angle(i,j)$ does not vary for vehicles at different positions, and thus, a fixed μ_{Angle} is sufficient for a specific surveillance scene. However, $Dist(i,j)$ varies much as a vehicle moves toward or away from the camera. As shown in Fig. 3, $Dist(i,j)$ is much larger when a vehicle is near the surveillance camera. More specifically, $Dist(i,j)$ is large when $v^{(i)}$ is small, and $Dist(i,j)$ is small when $v^{(i)}$ is large. Therefore, it is not appropriate to assume the same μ_{Dist} for vehicles at all positions. The solution is to observe the mean distance between headlight pairs $\mu_{\text{Dist}}^{(\text{Ref})}$ at a specific reference vertical coordinate $v^{(\text{Ref})}$. Afterward, when applying (7), the desired mean distance μ_{Dist} is obtained by adjusting $\mu_{\text{Dist}}^{(\text{Ref})}$ according to the current coordinate $v^{(i)}$ of the candidate pair.

The relations among the real-world coordinate system XYZ, the camera coordinate system $X_cY_cZ_c$, and the image plane coordinate system uv are discussed in [29]. From [29], we know that, for a point (X,Y) on the ground plane, the corresponding image plane coordinate (u,v) can be expressed by

$$u = f\frac{X}{Y\cos\phi + F} \tag{10}$$

$$v = f\frac{Y\sin\phi}{Y\cos\phi + F} \tag{11}$$

where ϕ is the tilt angle of the camera. In addition, F is the distance between the camera and the point where the line of

sight intersects the ground plane.The distance between the ith blob and the jth blob can be expressed as

$$Dist^2(i,j) = \left(u^{(j)} - u^{(i)}\right)^2 + \left(v^{(j)} - v^{(i)}\right)^2. \quad (12)$$

From (8), we know that

$$v^{(j)} - v^{(i)} = \left(u^{(j)} - u^{(i)}\right)\tan\psi. \quad (13)$$

By substituting $v^{(j)} - v^{(i)}$ in (12) using (13), we have

$$Dist^2(i,j) = \left(u^{(j)} - u^{(i)}\right)^2 (1 + \tan^2\psi). \quad (14)$$

Suppose that $(X^{(i)}, Y^{(i)})$ and $(X^{(j)}, Y^{(j)})$ are the corresponding ground plane coordinates for $(u^{(i)}, v^{(i)})$ and $(u^{(j)}, v^{(j)})$, respectively. Equation (12) can be expressed as

$$Dist^2(i,j) = \left(\frac{fX^{(j)}}{Y^{(j)}\cos\phi + F} - \frac{fX^{(i)}}{Y^{(i)}\cos\phi + F}\right)^2 (1 + \tan^2\psi). \quad (15)$$

Assume that the average width of the headlight pair of a vehicle is W. More specifically, $X^{(j)} - X^{(i)} = W$. In addition, we can assume that $Y^{(i)} = Y^{(j)}$ for the headlights of the same vehicle. Then, (15) can be rewritten as

$$Dist^2(i,j) = \left(\frac{fW}{Y^{(i)}\cos\phi + F}\right)^2 (1 + \tan^2\psi). \quad (16)$$

From (10) and (11), we can express $Y^{(i)}$ as

$$Y^{(i)} = \frac{Fv^{(i)}}{f\sin\phi - v^{(i)}\cos\phi}. \quad (17)$$

Substituting $Y^{(i)}$ in (16) using (17), then (16) becomes

$$Dist^2(i,j) = \left(\frac{fW}{\frac{Fv^{(i)}}{f\tan\phi - v^{(i)}} + F}\right)^2 (1 + \tan^2\psi). \quad (18)$$

It has been shown in [29] that $f\tan\phi = v^{(\text{vanish})}$. Here, $v^{(\text{vanish})}$ denotes the v coordinate of the vanishing point. Similarly, the distance between blob i_{Ref} and j_{Ref} can be obtained. Therefore, we can divide $Dist^2(i,j)$ by $Dist^2(i_{\text{Ref}}, j_{\text{Ref}})$ and get the following equation:

$$\frac{Dist^2(i,j)}{Dist^2(i_{\text{Ref}}, j_{\text{Ref}})} = \frac{\left(\frac{v^{(\text{Ref})}}{v^{(\text{vanish})} - v^{(\text{Ref})}} + 1\right)^2}{\left(\frac{v^{(i)}}{v^{(\text{vanish})} - v^{(i)}} + 1\right)^2}. \quad (19)$$

Thus, by exploring the camera geometry, we can obtain the desired μ_{Dist} for a candidate pair using

$$\mu_{\text{Dist}} = \frac{\left(1 + \frac{v^{(\text{Ref})}}{v^{(\text{vanish})} - v^{(\text{Ref})}}\right)}{\left(1 + \frac{v^{(i)}}{v^{(\text{vanish})} - v^{(i)}}\right)} \mu_{\text{Dist}}^{(\text{Ref})}. \quad (20)$$

As indicated by (7) and (9), we do not need to consider blobs that are far away from each other or forming angles that are too large. Therefore, the number of pairs that needs to be considered in the candidate list is significantly decreased. A likelihood value is computed for each remaining candidate pair according to (6). Then, the likelihood is sorted. The pair with the highest likelihood is confirmed as a paired headlight, and the paired blobs are deleted from the candidate list. All candidate pairs involving any of the paired blobs are also deleted from the candidate list. Then, the process is repeated until the candidate list is empty.

IV. Vehicle Tracking

In this section, we elaborate the tracking mechanism in the proposed system. The tracking framework is based on that described in [16]. First, we utilize a KF [30] to perform prediction. Afterward, we obtain reliable measurements to update the KF. Under normal conditions, segmented foreground objects can be used as measurements to update the filters. However, when segmentation errors or temporal occlusions are detected, we cannot rely on the segmented foreground as measurements. Therefore, dynamic particle sampling [16] is performed in the case of segmentation errors or temporal occlusions. The algorithm is efficient, because particles are generated only when necessary and the sampling parameters are dynamically determined by the innovation covariance of the KF at the previous time instance. Such design avoids the degeneracy problem of a particle filter (PF) and the need for resampling. In addition, a smaller number of particles are required to achieve the tracking accuracy, compared with classical PF. While we use particle sampling to provide reasonable measurement candidates, the mathematical tractability and closed-form solutions provided by KF do not need to be sacrificed. Then, we utilize an enhanced version of probabilistic data association [18] to associate the measurements with each tracking target for filter update. For the design of the KF, the system state at time instance k is defined as $\mathbf{x}_k = [u_k v_k \dot{u}_k \dot{v}_k a_k b_k]^T$, where (u_k, v_k) denotes the centroid of the vehicle in the image plane. The state elements $\dot{u}_k$ and $\dot{v}_k$ denote the velocities of the vehicle on the image plane, i.e., the displacements in unit time, of u and v directions, respectively. The size parameters (a_k, b_k) are the lengths of the major axis and the minor axis of the ellipse fitted on the tracked vehicle. The measurement is defined as $\mathbf{y}_k = [u_k v_k a_k b_k]^T$. The vehicles are initialized when the amount of accumulated foreground pixels at the initialization gate reaches a threshold. Therefore, for large vehicles, we track its head, instead of the entire vehicle. Highway surveillance cameras are usually set facing the traffic flow, as shown in Fig. 4. Under such setting of surveillance cameras, we could observe that, for a vehicle with constant speed, the displacement of the vehicle gets larger on the image plane as the vehicle moves toward the camera. In addition, the displacement of the vehicle gets smaller on the image plane as the vehicle moves away from the camera. Therefore, a fixed-system state transition matrix is insufficient for system state prediction in the KF. The main improvement on tracking is that the system state transition model of the KF is adaptive and time varying in this work.

To design a time-varying-system state transition matrix that could capture the characteristics of vehicle movements on the

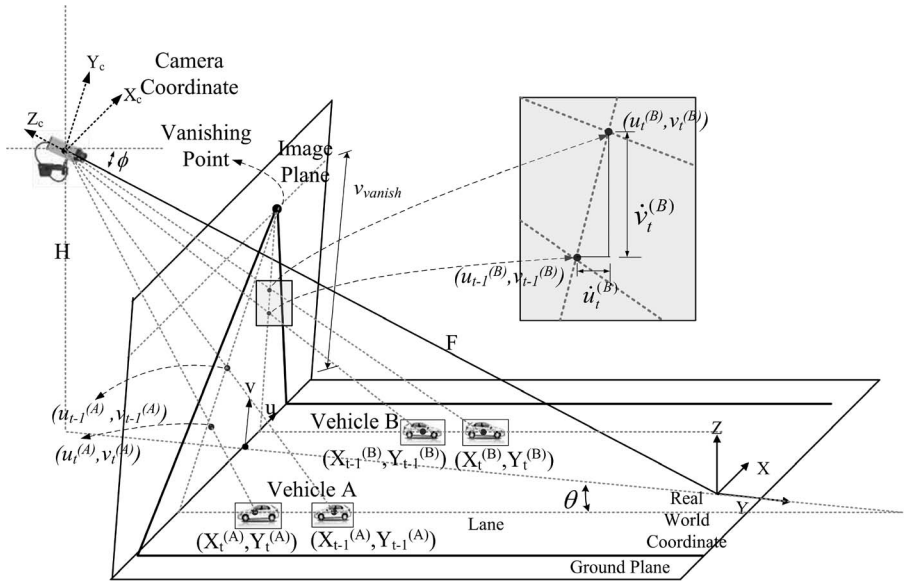

Fig. 4. Traffic surveillance camera and coordinate systems.

image plane, we need to explore the geometry of the camera and the roadway. As explained in Section III, the relation between a point (X, Y) on the ground plane and the corresponding point (u, v) on the image plane can be expressed as (10) and (11). We could assume that, on the ground plane, the vehicle velocities remain the same from time instance $t-1$ to time instance t, i.e., $\dot{X}_t = \dot{X}_{t-1}$ and $\dot{Y}_t = \dot{Y}_{t-1}$. Furthermore, the relations between (u_t, v_t) and (u_{t-1}, v_{t-1}) can be written as

$$u_t = u_{t-1} + \dot{u}_{t-1}, \qquad v_t = v_{t-1} + \dot{v}_{t-1}. \tag{21}$$

Under the aforementioned assumptions, it has been established in [31] that the predicted displacements $\dot{u}_t$ and $\dot{v}_t$ can be expressed by v_{t-1}, $\dot{u}_{t-1}$, $\dot{v}_{t-1}$, and $v^{(\text{vanish})}$, as shown in

$$\dot{u}_t = \left(1 - \frac{\dot{v}_{t-1}}{v^{(\text{vanish})} - v_{t-1}}\right)^2 \dot{u}_{t-1}$$

$$\dot{v}_t = \left(1 - \frac{\dot{v}_{t-1}}{v^{(\text{vanish})} - v_{t-1}}\right)^2 \dot{v}_{t-1}. \tag{22}$$

That is, we can predict the displacement of a vehicle on the image plane at time instance t based on the position and the velocity of the vehicle at time instance $t-1$. If the origin of the image plane is defined as shown in Fig. 4, we have $v^{(\text{vanish})} - v_{t-1} > 0$. If the vehicle is moving toward the camera as vehicle A in Fig. 4 does, then $\dot{v}_{t-1}^{(A)} = v_t^{(A)} - v_{t-1}^{(A)}$ would be smaller than zero. In this case, we could observe from (22) that $\dot{u}_t^{(A)} > \dot{u}_{t-1}^{(A)}$ and $\dot{v}_t^{(A)} > \dot{v}_{t-1}^{(A)}$. This observation matches the fact that the displacements $\dot{u}_t^{(A)}$ and $\dot{v}_t^{(A)}$ become increasingly larger in the image plane as t increases, when the vehicle moves toward the camera. On the other hand, if $\dot{v}_{t-1} > 0$, i.e., the vehicle is moving away from the camera as vehicle B in Fig. 4 does, then we have $\dot{u}_t^{(B)} < \dot{u}_{t-1}^{(B)}$ and $\dot{v}_t^{(B)} < \dot{v}_{t-1}^{(B)}$. That is, the displacements $\dot{u}_t^{(B)}$ and $\dot{v}_t^{(B)}$ become increasingly smaller in the image plane as t increases. Therefore, the time-varying-system state transition matrix can be defined as

$$F_t = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & \gamma_t & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_t & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta_t & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta_t \end{bmatrix} \tag{23}$$

$$\gamma_t = \left(1 - \frac{\dot{v}_{t-1}}{v_{\text{vanish}} - v_{t-1}}\right)^2 \tag{24}$$

$$\delta_t = \frac{\left(1 + \frac{v_{t-1}}{v^{(\text{vanish})} - v_{t-1}}\right)}{\left(1 + \frac{v_t}{v^{(\text{vanish})} - v_t}\right)}. \tag{25}$$

V. Estimation of Traffic Parameters Via Regression

If the self-diagnosis process determines that the current traffic condition is not suitable for performing tracking, the proposed system would estimate the traffic parameters using a mechanism based on regression. The basis of traffic parameter estimation via regression lies in the relationship between the observed feature x and the number of vehicles y passing a detection zone R in a fixed time duration ΔT. The observed features are the cumulated amount of moving edges (MEs) [32], as described in

$$x = \sum_{\Delta T} \sum_{(u,v) \in R} ME(u, v) \tag{26}$$

and the variation of color histogram, as described in

$$x = \sum_{\Delta T} \sum_{\text{all bins } k} \left| Hist_f(k) - Hist_{f+1}(k) \right|. \tag{27}$$

In (27), $Hist_f(k)$ denotes the value of the kth bin in the color histogram at frame f computed over the detection zone R.

We use the least square error method to find the second-order polynomial regression function [33]

$$y = ax^2 + bx + c. \tag{28}$$

Suppose that we have n pairs of points (x_i, y_i), $\mathrm{i} = 1, 2, \ldots, n$. The squared error is defined as

$$\mathrm{E} = \sum_{i=1}^{n} \left[y_i - \left(ax_i^2 + bx_i + c \right) \right]^2. \tag{29}$$

To minimize the squared error E, take the partial derivative of E with respect to coefficients a, b, and c, respectively. Letting the derivatives to zeros yields the following equations:

$$\left(\sum_{i=1}^{n} x_i^2 \right) a + \left(\sum_{i=1}^{n} x_i \right) b + nc = \sum_{i=1}^{n} y_i \tag{30}$$

$$\left(\sum_{i=1}^{n} x_i^3 \right) a + \left(\sum_{i=1}^{n} x_i^2 \right) b + \left(\sum_{i=1}^{n} x_i \right) c = \sum_{i=1}^{n} x_i y_i \tag{31}$$

$$\left(\sum_{i=1}^{n} x_i^4 \right) a + \left(\sum_{i=1}^{n} x_i^3 \right) b + \left(\sum_{i=1}^{n} x_i^2 \right) c = \sum_{i=1}^{n} x_i^2 y_i. \tag{32}$$

According to Cramer's rule, we can obtain a, b, and c via the following equations:

$$a = \frac{D_1}{\Delta} \quad b = \frac{D_2}{\Delta} \quad c = \frac{D_3}{\Delta} \tag{33}$$

where

$$\Delta = \begin{vmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i & n \\ \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^4 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 \end{vmatrix} \tag{34}$$

$$D_1 = \begin{vmatrix} \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} x_i & n \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^2 y_i & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 \end{vmatrix} \tag{35}$$

$$D_2 = \begin{vmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} y_i & n \\ \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^4 & \sum_{i=1}^{n} x_i^2 y_i & \sum_{i=1}^{n} x_i^2 \end{vmatrix} \tag{36}$$

$$D_3 = \begin{vmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} x_i^4 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 y_i \end{vmatrix}. \tag{37}$$

Equations (34)–(37) yield a quadratic function. However, as x increases, y would increase first and drop afterward due to the shape of the parabolic curve. The parabola has a maximum value $(4ac - b^2)/4a$ at $x = -b/2a$ and then starts to drop when $x > -b/2a$. To match the fact that the number of vehicles should not drop as the amount of cumulated ME or variation of color histogram increases, we further constrain the regression function using the following equation:

$$y = \begin{cases} \frac{4ac - b^2}{4a}, & x \geq -\frac{b}{2a} \\ ax^2 + bx + c, & x < -\frac{b}{2a}. \end{cases} \tag{38}$$

The sample data that the system relies on to obtain the regression polynomial can be obtained in two different ways. The first way is fully automatic, and the second way is semiautomatic, involving human input. The fully automatic method collects the sample data using an existing vehicle counting method when the congestion condition is not severe. In Section VI, we would compare the fully automatic methods that use VIL and entropy method (EM) to collect the sample data for regression. For semiautomatic sample data collection, human input is involved to aid the system for better estimation accuracy. It is predicable that, when the sample data for regression are semiautomatically obtained, the system would demonstrate enhanced accuracy and extra robustness against segmentation errors of the foreground objects.

VI. Experimental Results

The experimental results are demonstrated and explained in this section. We test the proposed system on various traffic surveillance scenes obtained from the database of Taiwan Area National Freeway Bureau, Ministry of Transportation and Communications. Both daytime and nighttime surveillance scenes under different traffic conditions and with different camera angles are considered. The frame rate of the videos is 25 frame/s. Fig. 5 shows some snapshots of the traffic surveillance sequences. Notice that some surveillance sequences are under serious congestion conditions, and the effects of shadows are severe. The region of interest is the near scene under the solid horizontal line shown in Fig. 5. The region between the dotted lines is the detection zone for traffic flow estimation. For tracking, we need to specify the position of the solid horizontal line to determine the Region of Interest (ROI) and whether it is the entrance or exit to determine the direction of the traffic. For traffic flow estimation, we need to specify the dotted horizontal lines to determine the detection zone. The region of interest does not cover the distant view in the scene due to the following reasons: First, the resolution in the distant view is low. Second, the lanes are curved in the surveillance scenes, but we apply linear lane models here.

Fig. 6 shows the thresholding and headlight pairing results of some nighttime scenes. The first row of Fig. 6 displays the original nighttime frames with different gray-level histograms. The second and third rows of Fig. 6 are the histograms and backward accumulated histograms of the images in the first row, respectively. The vertical lines indicate the dynamic thresholds selected by the system. We can observe that there is no

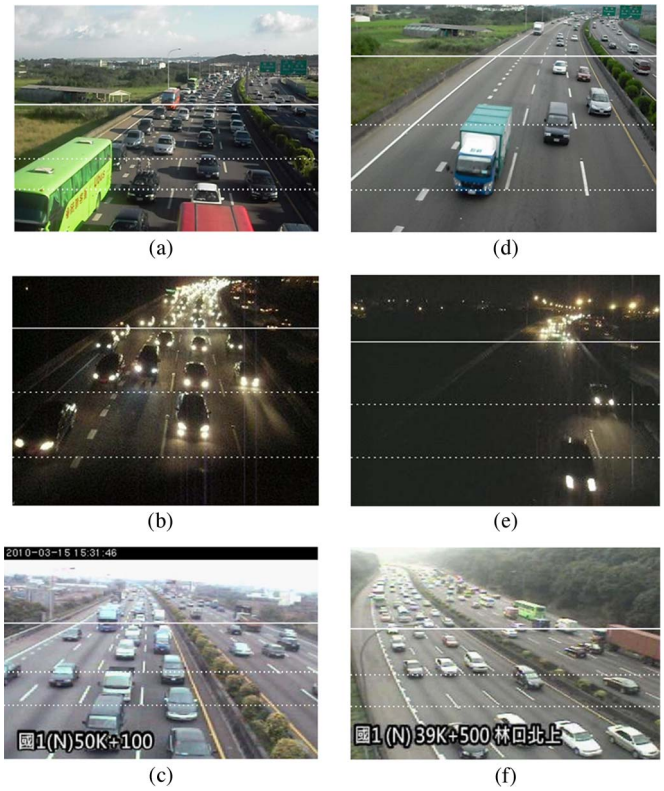

Fig. 5. Experimental sequences. (a) Sequence A. (b) Sequence B. (c) Sequence C. (d) Sequence D. (e) Sequence E. (f) Sequence F.

absolute relationship between the desired threshold and the average intensity of each image frame. The method of dynamic threshold selection via backward accumulated histograms is able to find the appropriate threshold for each frame according to its intensity distribution. To provide quantitative evaluations for thresholding, we have manually marked the ground truth of the headlights in the night surveillance scenes for 50 frames. In addition, we use these marked ground truth pixels to compute the average detection rate and the average number of false alarm pixels per frame. The average detection rate is 0.979, and the average number of false alarm pixels per frame is 307.72. Note that the detection rate is defined as the number of correctly detected pixels divided by the number of ground truth pixels. After thresholding, we use erosion, followed by dilation, to remove the tiny noises first and then to restore the shapes of the detected lighting objects. After thresholding and morphological operations, the resulting lighting objects include headlights, auxiliary lights, reflections of lights of the vehicles, and some lane markings, as shown in the fourth row of Fig. 6. The proposed headlight-pairing algorithm can successfully eliminate the nonheadlight blobs and correctly find the corresponding pairing headlight for each headlight blob.

Table I lists the detection and tracking accuracy under smooth traffic conditions. The statistics are collected in a 30-min interval from both daytime and nighttime videos. Comparisons of the detection accuracy among various nighttime methods demonstrate the effectiveness of the proposed headlight-pairing algorithm. The tracking accuracy demonstrates the advantage of the time-varying state transition matrix over the static state transition matrix for prediction. The tracking accuracy is defined as the number of correctly detected and tracked vehicles divided by the number of correctly detected vehicles. For template matching and rule-based headlight-pairing methods, we do not consider the subsequent tracking accuracy based on these two detection methods.

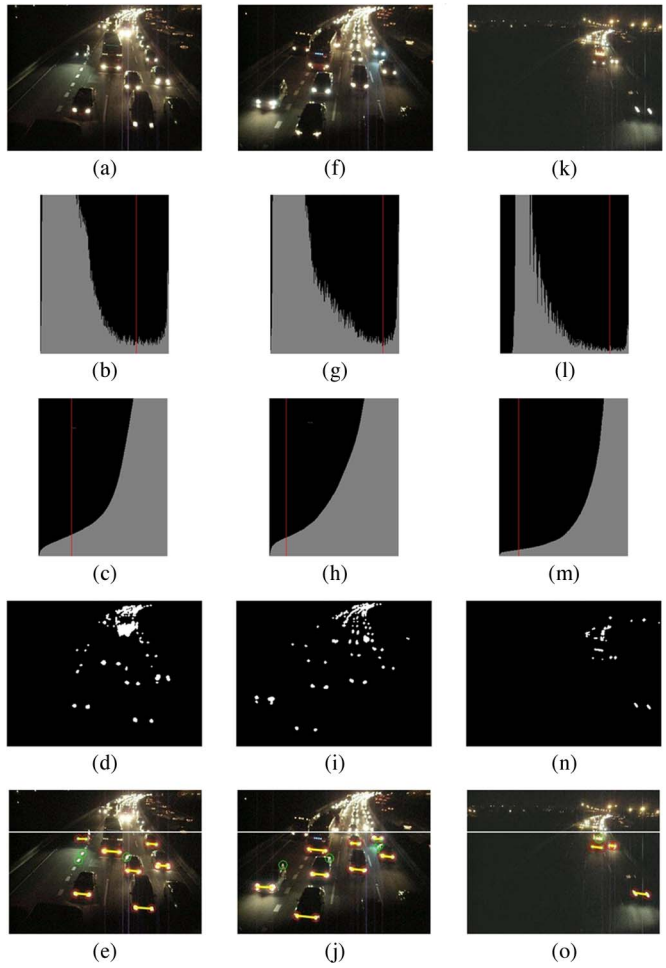

Fig. 6. Thresholding and headlight pairing for nighttime surveillance.

TABLE I
DETECTION AND TRACKING ACCURACY

	Vehicle Detection Method	Detection Precision	Detection Recall	Tracking Accuracy	
				Static State Transition	Time-varying State Transition
Daytime	Background modeling with shadow removal	98.56%	98.77%	99.12%	99.37%
Nighttime	Template matching [5]	96.32%	96.35%	N/A	N/A
	Cascade classifiers [9]	97.29%	97.10%		
	Proposed method	98.02%	98.20%	98.96%	99.29%

Fig. 7 compares the estimated flow accuracy for congested sequences using different features for regression. The estimated flow accuracy is measured by the mean absolute error between the estimated flow and the ground truth flow within a 5-min

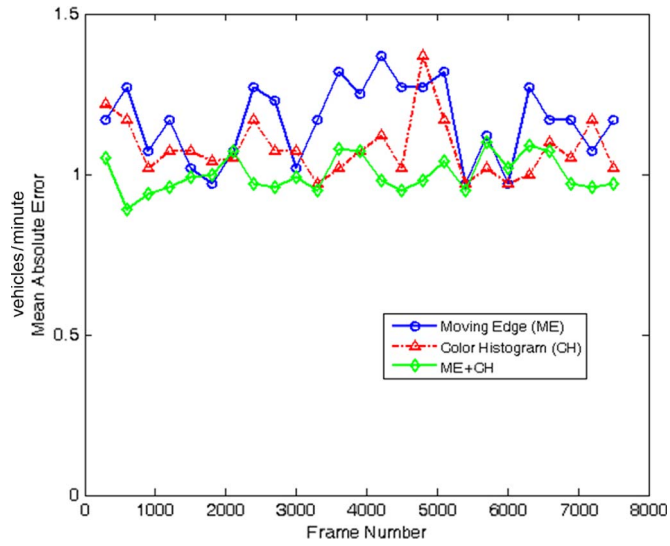

Fig. 7. Mean absolute error of flow estimation on congested sequences using different features.

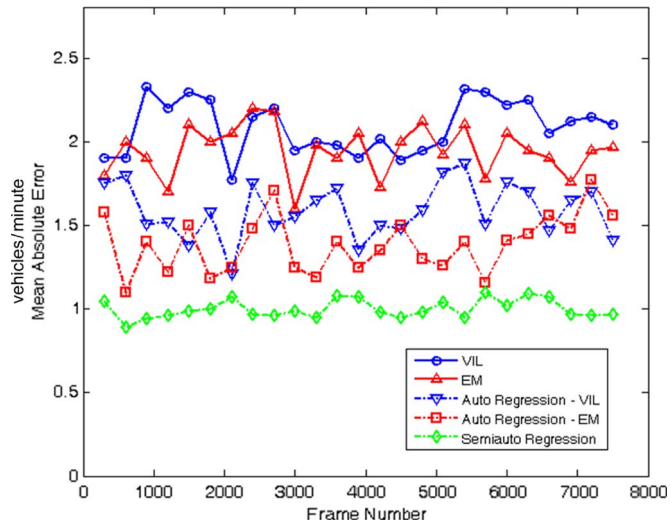

Fig. 8. Mean absolute error of flow estimation on congested sequences using different methods.

interval for sequences A and C. The unit of the flow is the number of vehicles per minute. For sequence A, the statistics are collected from all the four lanes. However, for sequence C, the statistics are collected from the congested third and fourth lanes only. In this experiment, we obtain data samples for regression semiautomatically. Each data sample for regression is obtained from a fixed time duration $\Delta T = 300$ frames. Since ME + CH yields slightly better results, it is used for the rest of the experiments.

Fig. 8 compares the estimated flow accuracy for congested sequences using different methods. The intervals for statistics collection are the same as those in Fig. 7. We compare the proposed method with the VIL [21] method and EM [22]. Because there are many large vehicles in sequence A, VIL would tend to count more vehicles than the ground truth. The performance of the EM is slightly better than that of the VIL. However, when the congestion condition is serious, the model in [22] cannot accurately capture the behavior of

TABLE II
TRAFFIC PARAMETER ESTIMATION

	Ground Truth Flow	Flow q	Occupancy	Manually Measured Speed (km/hr)	Space Mean Speed $\overline{s}$ (km/hr)
Sequence A	15	15.65	0.73	17.6	20.5
Sequence B	25	24	0.36	64.5	63.9
Sequence C	16	16.50	0.76	18.1	20.8
Sequence D	34	34	0.32	95.2	100.5
Sequence E	27	27	0.24	102.8	108.0
Sequence F	36	35	0.38	89.1	87.6

vehicles passing the detection zone. Constructing a regression polynomial helps the system not to overcount the number of vehicles when the detection zone is constantly occupied by vehicles. However, the traffic conditions during the training sample collection interval do have impact on the accuracy of estimation. If the training samples for regression polynomial construction are collected during time interval under severe congestion conditions, the accuracy of estimation would be deteriorated. As we can expect, if the training samples for regression are semiautomatically collected, the estimation error would be decreased.

Table II lists selected traffic parameter extraction results using the proposed system. The traffic flow q is defined as

$$q = \frac{N_{\text{car}}}{T_{\text{duration}}} \tag{39}$$

where N_{car} is the total number of vehicles passing through a specific detection zone in a given time duration T_{duration}. Occupancy defines the fraction of time that vehicles are located in a specific detection zone. T_{pass} is the time required for a vehicle to pass through the detection zone, i.e.,

$$Occupancy = \frac{T_{\text{pass}}}{T_{\text{duration}}}. \tag{40}$$

With $Occupancy$ and flow q, we could calculate the space mean speed $\overline{s}$ [34] as

$$\overline{s} = \frac{q(L+d)}{Occupancy} \tag{41}$$

where L and d represent the average length of vehicles and the length of the detection zone, respectively. The statistics in Table II are collected from the third lane of each surveillance video in Fig. 5 during a 5-min interval. Note that, for sequence A and sequence C, the estimated flow is obtained via regression, and therefore, it is possible to obtain noninteger flow parameters in a 1-min interval.

Fig. 9 shows the averaged traffic flow accuracy for all the four lanes in surveillance sequence C within a 5-min interval. This experiment takes into account both smooth and congested traffic conditions, because the traffic conditions of the first two lanes are smooth, and the third and fourth lanes are congested in surveillance sequence C. If self-diagnosis is not performed, the tracking algorithm performs poorly under seriously congested

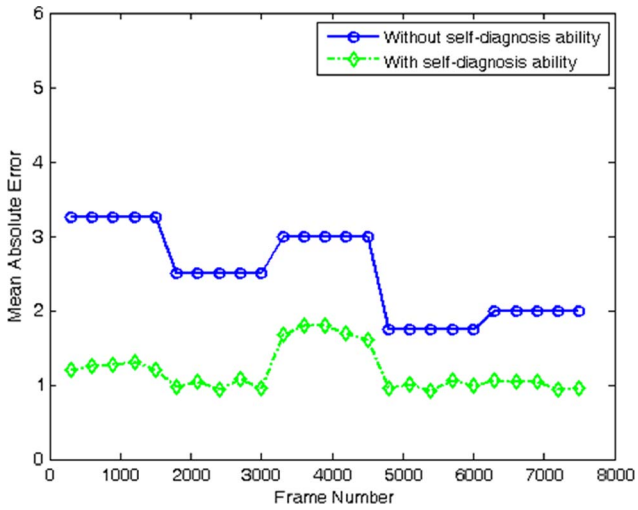

Fig. 9. Mean absolute error of traffic flow for all the four lanes in surveillance sequence C.

conditions, as expected. Therefore, the overall flow accuracy computed from tracking results would be deteriorated. With the self-diagnosis ability, the system performs regression when tracking is unreliable. More specifically, tracking is performed on the first two lanes, and regression is performed on the third and fourth congested lanes. As a result, the overall accuracy is significantly improved.

VII. Conclusion

For an intelligent traffic surveillance system, it is important to adapt the system to different surveillance scenes and traffic conditions. In this paper, we have proposed a highway intelligent surveillance system with self-diagnosis abilities and designed effective solutions for different conditions. One of the main contributions of this work includes a likelihood-based headlight-pairing mechanism for vehicle detection under nighttime surveillance. By analyzing camera geometry, we could obtain the distances between headlight pairs at different positions without prior assumptions of many camera parameters. Another important issue is to balance between abundance of acquired information and robustness. Performing tracking would be preferred when traffic is smooth to acquire useful information for subsequent event analysis modules. However, under severe congestion conditions, it is better to obtain traffic parameters by estimation, because tracking results would be unreliable under such circumstances. When performing tracking, we incorporate an efficient tracking framework that utilizes Kalman filtering for prediction, adaptive particle sampling for candidate measurement selection, and enhanced probabilistic data association for filter update. The main improvement in the tracking framework is that we utilize a time-varying adaptive system state transition matrix in the KF for better prediction in highway traffic surveillance scenes. For traffic parameter estimation, we propose a regression-based mechanism. The experimental results have shown that the self-diagnosis ability and the modules designed for the system make the proposed system robust and reliable.

References

[1] K. W. Dickinson and R. C. Waterfall, "Video image processing for monitoring road traffic," in *Proc. IEE Int. Conf. Road Traffic Data Collect.*, Dec. 5–7, 1984, pp. 105–109.
[2] M. Kilger, "Video-based traffic monitoring," in *Proc. Int. Conf. Image Process. Appl.*, Maastricht, The Netherlands, Apr. 7–9, 1992, pp. 89–92.
[3] T. Semertzidis, K. Dimitropoulos, A. Koutsia, and N. Grammalidis, "Video sensor network for real-time traffic monitoring and surveillance," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 103–112, Jun. 2010.
[4] 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.
[5] 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.
[6] Y. L. Chen, C. T. Lin, C. J. Fan, C. M. Hsieh, and B. F. Wu, "Vision-based nighttime vehicle detection and range estimation for driver assistance," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2008, pp. 2988–2993.
[7] R. O'Malley, E. Jones, and M. Glavin, "Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 453–462, Jun. 2010.
[8] K. Robert, "Night-time traffic surveillance: A robust framework for multi-vehicle detection, classification and tracking," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveillance*, 2009, pp. 1–6.
[9] W. Wang, C. Shen, J. Zhang, and S. Paisitkriangkrai, "A two-layer night-time vehicle detector," in *Proc. Digit. Image Comput., Tech. Appl.*, 2009, pp. 162–167.
[10] D. J. Dailey, F. W. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 98–107, Jun. 2000.
[11] 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.
[12] S. Atev, G. Miller, and N. P. Papanikolopoulos, "Clustering of vehicle trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 30, pp. 647–657, Sep. 2010.
[13] H. Veeraraghavan and N. P. Papanikolopoulos, "Learning to recognize video-based spatiotemporal events," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 628–638, Dec. 2009.
[14] A. Yoneyama, Y. C. Hung, and C. C. J. Kuo, "Robust vehicle and traffic information extraction for highway surveillance," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 2305–2321, Jan. 2005.
[15] D. R. Magee, "Tracking multiple vehicles using foreground, background and motion models," *Image Vis. Comput.*, vol. 22, no. 2, pp. 143–155, Feb. 2004.
[16] H. Y. Cheng and J. N. Hwang, "Adaptive particle sampling and adaptive appearance for multiple video object tracking," *Signal Process.*, vol. 89, no. 9, pp. 1844–1849, Sep. 2009.
[17] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 267–276, Jun. 2010.
[18] H. Y. Cheng and J. N. Hwang, "Multiple-target tracking for crossroad traffic utilizing modified probabilistic data association," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2007, pp. I-921–I-924.
[19] M. Fathy and M. Y. Siyal, "A window-based image processing technique for quantitative and qualitative analysis of road traffic parameters," *IEEE Trans. Veh. Technol.*, vol. 47, no. 4, pp. 1342–1349, Nov. 1998.
[20] M. Y. Siyal and M. Fathy, "Image processing techniques for real-time qualitative road traffic data analysis," *Real-Time Imag.*, vol. 5, no. 4, pp. 271–278, Aug. 1999.
[21] E. Viarani, "Extraction of traffic information from images at DEIS," in *Proc. Int. Conf. Image Anal. Process.*, 1999, pp. 1073–1076.
[22] W. L. Hsu, H. Y. M. 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.
[23] M. Fathy and M. Y. Siyal, "A neural-vision based approach to measure traffic queue parameters in real-time," *Pattern Recognit. Lett.*, vol. 20, no. 8, pp. 761–770, Aug. 1999.
[24] D. Lee and Y. Park, "Measurement of traffic parameters in image sequence using spatio-temporal information," *Meas. Sci. Technol.*, vol. 19, no. 11, pp. 1–9, Nov. 2008.
[25] N. Martel-Brisson and A. Zaccarin, "Learning and removing cast shadows through a multidistribution approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1133–1146, Jul. 2007.
[26] A. H. S. Lai and N. H. C Yung, "Lane detection by orientation and length discrimination," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 4, pp. 539–548, Aug. 2000.

[27] C. Harris and M. J. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–152.
[28] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ: Wiley, 2001.
[29] T. N. Schoepflin and D. J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 2, pp. 90–98, Jun. 2003.
[30] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME—J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
[31] 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.
[32] C. Kim and J. N. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 2, pp. 122–129, Feb. 2002.
[33] J. D. Gergonne, "The application of the method of least squares to the interpolation of sequences," *Historia Math.*, vol. 1, no. 4, pp. 439–447, Nov. 1974.
[34] N. H. Gartner, C. J. Messer, and A. Rathi, "Traffic flow theory: A state of art report," Transp. Res. Board, Washington, DC, 1992.

Hsu-Yung Cheng (M'08) was born in Taipei, Taiwan. She received the B.S. degree in computer science and information engineering and the M.S. degree from National Chiao-Tung University, Hsinchu, Taiwan, in 2000 and 2002, respectively, and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, in 2008.

In 2008, she joined the Department of Computer Science and Information Engineering, National Central University, Chung-li, Taiwan, as an Assistant Professor. Her research interests include image and video analysis and intelligent systems.

Shih-Han Hsu was born in Taipei, Taiwan. He received the B.S. degree in mechanical engineering from Chung-Yuan Christian University, Jhongli City, Taiwan, in 2006 and the M.S. degree in computer science and information engineering from Yuan-Ze University, Jhongli City, in 2008. He is currently working toward the Ph.D. degree in computer science and information engineering from the Department of Computer Science and Information Engineering, National Central University, Chung-li, Taiwan.

His research interests include image processing and machine intelligence.