Open access • Journal Article • DOI:10.1109/TIE.2013.2284131

A Relative-Discriminative-Histogram-of-Oriented-Gradients-Based Particle Filter Approach to Vehicle Occlusion Handling and Tracking — Source link

Bing-Fei Wu, Chih-Chung Kao, Cheng-Lung Jen, Yen-Feng Li ...+2 more authors

Institutions: National Chiao Tung University

Published on: 01 Aug 2014 - IEEE Transactions on Industrial Electronics (IEEE Industrial Electronics Society)

Topics: Video tracking, Histogram of oriented gradients, Particle filter and Vehicle tracking system

A Relative-Discriminative-Histogram-of-Oriented-Gradients-Based Particle Filter Approach to Vehicle Occlusion Handling and Tracking

Bing-Fei Wu, *Fellow, IEEE*, Chih-Chung Kao, *Student Member, IEEE*, Cheng-Lung Jen, *Student Member, IEEE*, Yen-Feng Li, Ying-Han Chen, and Jhy-Hong Juang

***Abstract*—This paper presents a relative discriminative histogram of oriented gradients (HOG) (RDHOG)-based particle filter (RDHOGPF) approach to traffic surveillance with occlusion handling. Based on the conventional HOG, an extension known as RDHOG is proposed, which enhances the descriptive ability of the central block and the surrounding blocks. RDHOGPF can be used to predict and update the positions of vehicles in continuous video sequences. RDHOG was integrated with the particle filter framework in order to improve the tracking robustness and accuracy. To resolve multiobject tracking problems, a partial occlusion handling approach is addressed, based on the reduction of the particle weights within the occluded region. Using the proposed procedure, the predicted trajectory is closer to that of the real rigid body. The proposed RDHOGPF can determine the target by using the feature descriptor correctly, and it overcomes the drift problem by updating in low-contrast and very bright situations. An empirical evaluation is performed inside a tunnel and on a real road. The test videos include low viewing angles in the tunnel, low-contrast and bright situations, and partial and full occlusions. The experimental results demonstrate that the detection ratio and precision of RDHOGPF both exceed 90%.**

***Index Terms*—Histogram of oriented gradients (HOG), particle filter, vehicle tracking.**

I. Introduction

Traffic monitoring is an important issue during traffic management. Therefore, many detection methods have been proposed in the literature, which use monocular cameras, stereo cameras, and active sensors, such as radar and infrared. Due to falling costs and the increased efficiency of computational power, vision-based techniques have become popular for monitoring traffic scenes, and comprehensive up-to-date surveys are provided in [1] and [2]. However, it is difficult to locate and track targets precisely in low-contrast and excessively bright scenes. The bright object pairing method is feasible for vehicle tracking, and it is reliable in low-illumination situations [3]. A histogram extension method [4] was used to extract vehicular objects from highways and urban roads. Background subtraction, which obtains the foreground (moving) pixels by subtracting the current input image from a background image [5]–[10], is one of the most efficient methods for change detection, and it performs well when separating vehicles from the background, even with shadows and tree vibrations. In this paper, background subtraction is used to separate the vehicles from the background, while the relative discriminative histogram of oriented gradients (HOG) (RDHOG) method is applied for tracking vehicles.

After detection, visual tracking [30] is used to correlate vehicles that appear in successive frames. High-level event detection can be achieved using vehicle tracking. As well as detection, there have been many major advances in visual object tracking in recent decades. The most common methods rely on Bayesian filtering, such as Kalman filtering and particle filtering. Moreover, a recent review paper [11] summarizes the application of these filters to vision analysis. Traditionally, Kalman filters have been used to predict and update the states of linear and Gaussian models. Thus, head tracking [12] and target detection and following [13] have been performed based on Kalman filters. A particle filter [14], also known as the sequential Monte Carlo method or condensation algorithm, has been used for target localization in situations with large numbers of particles (samples). The particle filter method [15]–[24] has become very popular because of its ability to handle nonlinear and non-Gaussian models, as well as its robustness and flexibility. In [15], the contextual confidence of the measurement model was also considered, and nonoverlapping fragments were selected dynamically for likelihood measurements. Two operation modes have been proposed for the adaptive particle filter (APF) [16]. When the tracked vehicle is not occluded, APF uses a normal probability density function to generate a new set of particles. When the tracked vehicle is occluded, however, APF generates a new set of particles using a Normal-Rayleigh probability density function. Moreover, a motion-based model [17] was used to minimize the variance in the particle weights. In general, a first-order hidden Markov model is used during visual tracking, although a novel visual tracking framework that uses a high-order Monte Carlo Markov chain has been presented [18]. However, the computational complexity is increased as the order increases.

In [19], a projective transform was integrated into the particle filter framework. However, the parameters of transformation,

Manuscript received March 4, 2013; revised June 7, 2013; accepted July 18, 2013. Date of publication October 1, 2013; date of current version February 7, 2014. This work was supported by the National Science Council, Taiwan, under Grant NSC 100-2221-E-009-041. This work is specifically supported by “Aiming for the Top University Program” of the National Chiao Tung University and the Ministry of Education, Taiwan.

The authors are with the Institute of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: bwu@mail.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/TIE.2013.2284131

such as the height and the angle of the set camera, were difficult to obtain in each scene. A tracking algorithm based on a combination of a particle filter and the mean shift with a new transition model was presented in [20]. The multimodel anisotropic mean shift was integrated into the particle filter to reduce the number of particles, although the computational load was still high. In [21], a Kalman filter and a mean-shift tracker were integrated to predict a position more precisely. In addition to the Bhattacharyya coefficients used in color histograms, template matching is another method for particle weight calculation. A framework has been proposed that fuses salient points, blobs, and edges as features, while an object model adaptation was developed to facilitate more efficient operation [22]. Mei and Ling [23] proposed a visual tracking system that used a sparse representation set. The template was updated dynamically to capture changes in appearance, and nonnegativity constraints were enforced to filter out clutter that negatively resembled the tracking targets. A ten-layer annealed particle filter [24] was developed for visual tracking. In this framework, noise is adjusted adaptively by each layer, and the particles become tightly concentrated in the last layer. These techniques work well in normal light condition, but the low contrast in tunnels and areas with abnormal lighting conditions is seldom considered. Thus, the color histogram and image template are easily confused with the background image. Hence, the RDHOG method can be integrated into the particle framework to represent the target.

This paper proposes an RDHOG-based particle filter (RDHOGPF) for vehicle tracking with partial occlusion handling in a traffic surveillance application. RDHOGPF can be used to track passing vehicles more precisely. Unlike other methods that track objects using color histograms, the RDHOGPF tracking module locates and tracks vehicles accurately using RDHOG features. Based on HOG, the RDHOG features are the difference between the central block and the surrounding blocks. Thus, the weighting contributions of the features can be improved by combining the conventional HOG features and RDHOG features. In addition, each particle has two sample images: an image of the original size and another with a scaled width and height. Both sample images are resized into a standard pattern size of 32 × 32. If the sample image with a scaling factor has a larger weight after the likelihood calculation, the target size is updated with a scaling factor. Therefore, the target can be tracked and updated using the proper size. Occlusion handling is performed to maintain the tracked vehicle in a stable position when it is partially occluded.

Section II briefly introduces the workflow of the proposed approach. Section III describes the details of the proposed RDHOGPF for vehicle tracking with occlusion handling. Section IV presents our experimental evaluation of the proposed vehicle tracking method in particular scenarios. Finally, Section V concludes this paper.

II. Workflow

The proposed method combines RDHOGPF to improve the detection accuracy, according to the procedure shown in Fig. 1. Initially, background subtraction is used to detect and verify vehicles entering the region of interest (ROI). Background subtraction is a rapid approach for separating the vehicle from the background image. A mixture of Gaussian distributions is often used for modeling the color value of each pixel in an image over time.

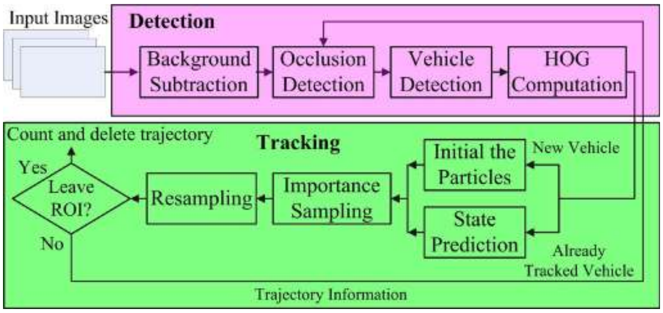


Fig. 1. Workflow of the proposed multivehicle detection and tracking method.

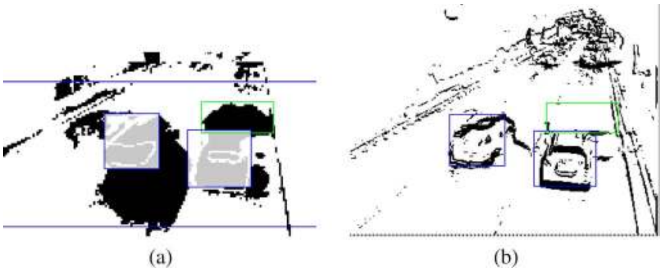


Fig. 2. Detection results for foreground segmentation and vehicle verification. (a) Foreground image. (b) Moving edge detection image.

After background subtraction, reflections in tunnels are easily recognized as the foreground pixels, particularly those caused by headlights in tunnels. Many previous studies have discussed shadow elimination, but the effect caused by headlight reflections in tunnels is also severe. It is difficult to distinguish the color of the headlight reflections from white vehicles. Therefore, an edge feature needs to be integrated into the system to segment the vehicle correctly. The edges of vehicles and the background are both extracted in the input image. To remove the edges caused by lane marks, moving edges are extracted by eliminating the edges from the background and the shadow. In Fig. 2(a), the false foreground pixels caused by the shadows and the headlights make it difficult for the system to identify the vehicle's position. Thus, we need to scan the horizontal edges in an upward direction to discriminate the real objects from the rigid body. The foreground pixels are drawn as gray in blue rectangles, which represent the segmented vehicles with moving edges. Otherwise, the green rectangle represents the foreground pixels attributable to the headlights. When the object is verified as a vehicle, the initial size of the rectangle is obtained for the vehicle. Using the verification procedure, the glare of the front light of coming vehicles can be recognized as a nonvehicle.

The trajectory is generated if the vehicle is detected more than twice in consecutive frames. The partial occlusions are detected and segmented based on the feedback tracking information. The tracked vehicle is represented and tracked using an RDHOG descriptor. The proposed vehicle tracking method

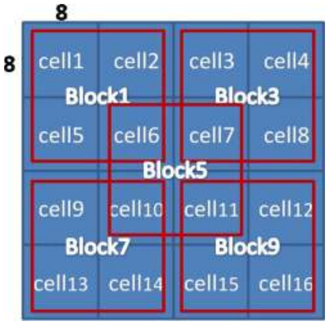


Fig. 3. Diagram of the cell and block used in the normalized image.

based on a particle filter can locate the target in the spatial space accurately and also obtain the correct size of the tracking vehicle as it passes.

III. RDHOGPF Vehicle Tracking Algorithm

Several important methods are introduced in this section for our proposed appearance-adaptive vehicle tracking framework, including the RDHOG computation, trajectory generation, a particle filter based on a two-scale RDHOG, appearance-adaptive template matching, and occlusion handling.

A. RDHOG Feature Computation

HOG is now a well-known descriptor used for object detection [26], [27]. The image gradient is computed by convolving the original image using a horizontal kernel $[-1\ 0\ 1]$ and a vertical gradient kernel $[-1\ 0\ 1]^{\mathrm{T}}$. In this paper, the sizes of all the target templates and sample images (particles) are normalized to 32 × 32. Next, the image is divided into equal cells of 8 × 8, which are represented as blue rectangles in Fig. 3.

In this paper, the orientation of the gradient is weighted by voting using nine bins spaced over 0°–180°. We take the magnitude of each pixel as the weight of the angle bin for any pixels present in the cell. We select four connected cells as a block. Therefore, the red rectangle in Fig. 3 contains nine blocks in the 32 × 32 image, and there are 36 bins in each block. These are combined with all blocks to produce a feature vector that serves as the vehicle descriptor. This is the basic HOG descriptor used to describe a target vehicle, and there are 324 HOG bins in each descriptor. HOG can be used to describe the shapes and magnitudes of the target image, but RDHOG is proposed to enhance the description of the target. In contrast to the conventional HOG bins, the relationships are considered between the central block and other surrounding blocks. The RDHOG bins indicate the difference between the central block and the surrounding blocks. Therefore, the difference between the HOG bins in the central block and those in the block around the central block is calculated using

$$RDb_j(i) = b_5(i) - b_j(i), \qquad j = 1 \sim 9, \quad j \neq 5 \tag{1}$$

where $b_j(i)$ is the ith bin of the jth block, $b_5(i)$ is the ith bin of the central block, and $RDb_j(i)$ is the ith RDHOG bin of the jth block. Therefore, there are 288 RDHOG bins in each descriptor. The total number of elements in the descriptor is 612 after combining HOG features and RDHOG features. Two scaling images are also calculated to track the vehicle at the correct size. If the vehicle drives in the y-direction in the real world, the appearance of the vehicle in the image will increase or decrease gradually. Therefore, in our approach, each particle has two samples: the original width (W) and height (H) of the tracked vehicle and the width (sW) and height (sH) with a scaling factor (this scaling factor depends on the testing scene). The vehicle becomes gradually smaller in the test scene, so the scaling factor is smaller than one. Next, both sample images with different image sizes are resized to 32 × 32, and HOG features are calculated. In addition, to reduce aliasing from the background, only the gradients in the foreground image are considered when calculating the HOG and RDHOG features. After the trajectory of the detected vehicle is generated, the descriptor of the vehicle is obtained for vehicle tracking.

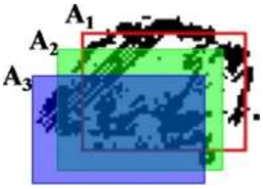


Fig. 4. Diagram of trajectory generation.

B. Trajectory Generation

The trajectory must be generated after detection to track a vehicle in an image sequence. The movements of the vehicles must be continuous if the frame rate is sufficiently high. The detected vehicles that appear in successive frames must overlap. Thus, the trajectory can be established when the vehicle is detected more than twice, and it overlaps in the consecutive frames. Thus, the correct detection result for a passing vehicle must include relationships for times t, $t-1$, and $t-2$. The detected regions overlap if these regions are related to the same vehicle. As shown in Fig. 4, A_1, A_2, and A_3 are the detected vehicles at times t, $t-1$, and $t-2$, respectively. a_1, a_2, and a_3 are the respective areas of A_1, A_2, and A_3, and $a_{1,2}$ and $a_{2,3}$ are the corresponding overlapping areas

$$a_i \triangleq \mathrm{area}(\mathrm{A}_1) \text{ and } a_2 \triangleq \mathrm{area}(\mathrm{A}_2) \tag{2}$$

$$a_{1,2} \triangleq \mathrm{area}(\mathrm{A}_1 \cap \mathrm{A}_2) \text{ and } a_{2,3} \triangleq \mathrm{area}(\mathrm{A}_2 \cap \mathrm{A}_3) \tag{3}$$

where area(A) is defined as the area of vehicle A. In our experiments, the trajectory is generated if the corresponding overlapping areas are larger than half the size of the respective areas a_1 and a_2.

C. Particle Filter

We assume a first-order Markov chain state-space model for Bayesian filtering. The particle filter is one of the most popular Bayesian filters used to estimate the posterior distributions of state variables. It comprises two main steps: prediction and updating. The state vector $\mathbf{x}_k$ is modeled as the position and the speed of the tracked vehicle in the image at discrete time k

$$\mathbf{x}_k = [x\ y\ \dot{x}\ \dot{y}]^T \tag{4}$$

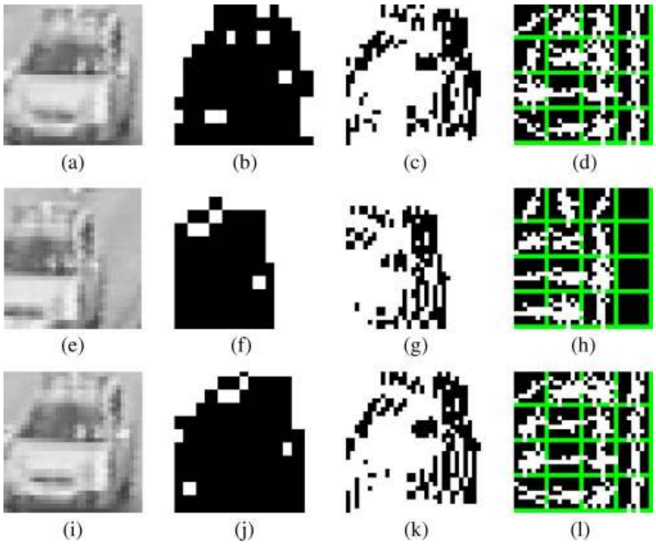


Fig. 5. Example of a HOG descriptor. (a) Target intensity image. (b) Target binary foreground image. (c) Target binary edge image. (d) Target HOG image. (e)–(h) Corresponding images of particle 1. (i)–(l) Corresponding images of particle 2.

where x and y represent the column and row position of the vehicle in the image, respectively, and $\dot{x}$ and $\dot{y}$ are the respective speeds of the vehicle in the subsequent frame.

Using the set of weighted particles S_k, the state estimate $\hat{\mathbf{x}}_k$ at time k can be the maximum *a posteriori* estimate

$$\hat{\mathbf{x}}_k = \arg\max_{S_k} \omega_k^{(j)} \tag{5}$$

where $\omega_k^{(j)}$ is the weight of the jth particle $\mathbf{x}_k^{(j)}$. Next, the samples are resampled to avoid a degeneration problem by deleting the particles with negligible weights and generating particles with equal weights from the high weight particles. Finally, the estimation results are refined and updated.

D. Likelihood Calculation

In this section, the state transition transits the states in the successive frames using white Gaussian noise, and the measurement model measures the similarity between the observations and the target. Unlike other methods that use color to represent the target, RDHOG is proposed to represent the tracked vehicle. RDHOG can address the shape property of the rigid body more clearly than the color histogram representation, and RDHOG also enhances the discrimination of the descriptor. To compute the similarity between the observation $\mathbf{y}_k$ (image within the bounding box) and the template $\mathbf{x}_k$, the observations are resized to the same size as the target template, 32 × 32. The HOG image and the binary edge image are shown in Fig. 5. The HOG features are extracted from the gradient image before the binary process. Using the edge image and foreground image, it is clear that the HOG features illustrated in Fig. 5 are different in the images of the target and the particles.

The similarity (or likelihood) $\mathrm{p}(\mathbf{y}_k|\mathbf{x}_k)$ that represents the similarity degree between the particle and the template is calculated based on the Bhattacharyya distance of the HOG feature. Using the HOG descriptor, the jth HOG weight of particle $h_k^{(j)}$ at time k can be computed based on the Bhattacharyya distance

TABLE I
PSEUDOCODE OF THE RESAMPLING ALGORITHM

Resample the samples and assign the new weight to each particle
$\left[\left\{x_{k+1}^j, w_{k+1}^j\right\}_{j=1}^N\right]$ = RESAMPLE $\left[\left\{x_k^j, w_k^j\right\}_{j=1}^N\right]$ ● Initialize the CDF : c_1=0 ● FOR i=2 to N $c_i = c_{i-1} + w_k^j$ END FOR ● Draw a straight point $u_i \sim \mathrm{U}[0, N^{-1}]$ ● FOR j=1 to N u_j=u_1+N^{-1}(j-1) WHILE u_j>c_i i=i+1 END WHILE Inherit the vector $x_k^j = x_k^i$ Inherit the weight $w_k^j = w_k^i$ END FOR

$$h_k^{(j)} = \sum_{i=0}^{N-1} \sqrt{d^{(j)}(i) d^T(i)} \tag{6}$$

where $d^{(j)}(i)$ and $d^T(i)$ are the ith elements of the HOG feature of the jth particle and target, respectively, and N, which is 324, is the total length of the HOG descriptor.

Next, the jth RDHOG weight of particle $\delta_k^{(j)}$ is calculated to enhance the discrimination of the target

$$\delta_k^{(j)} = \sum_{i=0}^{M-1} \left| RDb^{(j)}(i) - RDb^T(i) \right| \tag{7}$$

where $RDb^{(j)}(i)$ and $RDb^T(i)$ are the ith elements of the RDHOG features of the jth particle and target, respectively, and M, which is 288, is the total length of the RDHOG descriptor. Finally, the jth particle weight $\omega_k^{(j)}$ at time k can be computed using the following equation:

$$\omega_k^{(j)} = p\left(\mathbf{y}_k|\mathbf{x}_k^{(j)}\right) = h_k^{(j)} / \delta_k^{(j)}. \tag{8}$$

As $h_k^{(j)}$ increases, the similarity between the target and particle also increases. As $\delta_k^{(j)}$ becomes lower, the difference between the central block and surrounding blocks in the target and particles declines. Therefore, if $\omega_k^{(j)}$ is higher, there is greater similarity between the target and the particle. After obtaining the likelihood of the particles, we normalize the weights, so

$$\sum_{j=1}^{N} \omega_k^{(j)} = 1. \tag{9}$$

To resolve the degeneracy problem, the resampling procedure is performed to delete the lower weight particles and retain the strong particles. The pseudocode of the resampling algorithm is presented in Table I.

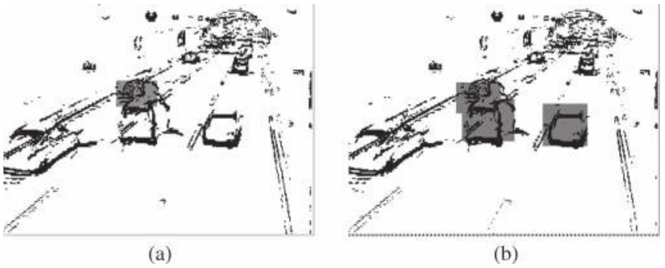


Fig. 6. New positions of the three trajectories estimated using RDHOGPF with partial occlusion handling for the edge image, which are masked with gray to avoid the generation of a new trajectory for the same vehicle in successive frames. (a) Position of the first trajectory. (b) Positions of the rest of the trajectories.

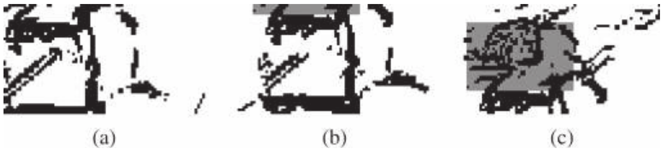


Fig. 7. Importance sampling example of the second trajectory during the RDHOGPF estimation. The (gray) masked region is the new region for the first trajectory. (a) Binary edge target image. (b), (c) Binary edge images (particles) with partially occluded regions.

E. Occlusion Handling

Most previous studies that addressed the problems of tracking have focused on resolving single object tracking. In most situations, however, two or more objects may appear simultaneously. In this paper, the older trajectory is estimated first, while the estimated position of the tracked vehicle is masked with gray, as shown in Fig. 6(a).

After updating the older trajectories, the newly generated trajectory estimates the new position using RDHOGPF. The masked region is drawn as gray based on the older trajectories, so some particles will contain the partial gray region shown in Fig. 7 when the sampling process is performed to calculate the likelihood.

The magnitudes are set to zero in the gray region using

$$M(u,v) = \begin{cases} 0, & \text{if } F(u,v) \sim= 1 \\ M(u,v), & \text{otherwise} \end{cases} \quad (10)$$

where u and v are the horizontal and vertical coordinates, respectively, and $M(u,v)$ and $F(u,v)$ are the respective magnitude and foreground values in the normalized 32 × 32 image. When the magnitude value is set to zero, the accumulated HOG feature $d^{(j)}(i)$ in (8) will decline, almost to zero in

$$d(i) = 0, \qquad \text{if } M(u,v) = 0 \quad (11)$$

where u and v are the horizontal and vertical coordinates in a cell, respectively. Therefore, the occluded regions will lead to decreases in the weights in (8). Thus, the estimation of the second trajectory will yield the correct position by evaluating the weight affected least by the other trajectory. Finally, the results of the estimation of three trajectories are exhibited in Fig. 6(b). Partial occlusion still occurs, but the trajectory can handle this situation and tracks the vehicles well. Moreover, the region in the estimation result is masked with a gray color in the edge and the foreground image, so the vehicle detection procedure will not detect and generate a new trajectory for the same vehicle in the subsequent frame.

TABLE II
EXECUTION TIME FOR EACH MODULE

Module	Average time (ms)	Standard deviation (ms)
Foreground extraction	0.745	0.110
Morphological operation	0.309	0.108
Obtain edge and the RDHOG features	6.826	0.509
Connected component	0.940	0.253
Vehicle verification	0.086	0.021
Trajectory generation	0.093	0.019
Particle filter tracking	12.374	9.624
Draw and display the tracking result	1.563	0.051
Total	22.936	10.695

IV. Experimental Results

This section addresses the performance evaluation results for the proposed RDHOGPF method of vehicle tracking using several traffic surveillance videos. Traffic video 1 was a continuous 1-h video recorded inside a tunnel in low-contrast conditions, while traffic video 2 was a continuous half-hour video in excessively bright light conditions with low contrast. The difficulties inherent in these conditions were as follows. First, because of the height limitations in the tunnel, vehicles were sometimes partially occluded when they entered the detection zone within the tunnel. Second, low-contrast and excessively bright scenes led to rough target representations. However, the proposed RDHOGPF approach handled these situations and maintained its performance. Our approach was implemented in C++ using the Microsoft Foundation Class Library. A highly detailed analysis of the computational time requirements is provided in Table II. Table II illustrates that most of the computational time was attributed to the particle filter tracking module because of the likelihood computation and the particle weight normalization. Because of the existing trajectories, the standard deviation of the particle filter tracking execution time was higher than that of the other modules. These scenarios were all tested using a Windows XP platform using an Intel core i5 with a 3.3-GHz central processing unit and 3 GB of RAM, where the test image size was 320 × 240.

In the quantitative evaluation of the vehicle tracking performance, we used the detection ratio (Recall) and precision ratio (Precision), which are used widely to evaluate the performance during information retrieval. These measures are defined as

$$\text{Recall} = T_p/(T_p + F_n) \quad (12)$$

$$\text{Precision} = T_p/(T_p + F_p) \quad (13)$$

where T_p (true positives) represents the number of correctly detected vehicles, F_p (false positives) represents the number of

TABLE III
DETECTION RESULTS IN TRAFFIC VIDEO 1 (1 h)

Lane	Def.	TSHT [25]	Annealing PF [24]	APF [16]	Contour PF [28]	RDHOGPF
Left Lane	*Tp*	1174	1321	1437	1550	1555
	Fn	483	336	220	107	102
	Fp	117	138	256	118	110
	Recall	70.85	79.72	86.72	**95.54**	93.84
	Precision	90.94	90.54	84.88	92.93	**93.39**
Middle Lane	*Tp*	1238	969	1300	1409	1440
	Fn	264	533	202	93	62
	Fp	105	71	222	39	47
	Recall	82.42	64.51	86.55	93.81	**95.87**
	Precision	92.18	93.17	85.41	**97.31**	96.84
Right Lane	*Tp*	1054	1294	1134	1406	1406
	Fn	481	241	401	129	129
	Fp	102	212	145	70	62
	Recall	68.66	84.30	73.88	**91.60**	**91.60**
	Precision	91.18	85.92	88.66	95.26	**95.78**
Total	*Tp*	3466	3584	3871	4365	4401
	Fn	1228	1110	843	329	293
	Fp	324	421	623	227	219
	Recall	73.84	76.35	82.47	93.37	**93.76**
	Precision	91.45	89.49	86.14	95.05	**95.26**

falsely detected vehicles, and F_n (false negatives) is the number of missing vehicles. We compared our RDHOGPF method with an APF [16], an annealing PF [24], a two-stage hybrid tracker (TSHT) [25], and a tracker that fused the color and contour features [28]. These approaches all deliver good performance according to the literature. TSHT is a hybrid tracker that combines a particle filter with mean shift, the annealing PF generates and resamples the particles in different layers with scalable variance, and APF handles occlusion using a Rayleigh distribution. The system presented in [28] performed well. However, its performance was still worse than that of RDHOGPF because of the latter's ability to cope with noise and poor lighting environments. RDHOGPF delivered a better overall performance than the other approaches, regardless of the lane to which it was applied or the performance measure selected.

Table III shows the results with each approach during vehicle detection and tracking using traffic surveillance video 1. RDHOGPF delivered a Recall of 93.75% and a Precision of 95.26%. Thus, our proposed approach could detect and track the passing vehicles correctly through the end of ROI. Lower values of F_n and F_p indicated lower numbers of missed vehicles and falsely detected vehicles, respectively. The parameters for each lane were adjusted for the comparison of the methods because the slopes were different in each lane. For TSHT, the standard deviation of the position of x and y was set to 6, the standard deviation of the size scale was set to 0.01, ε was 1, k_{p} was 15, and the particle number was 30. For annealing PF, the layer number was set to 3, P_0 is 6, the alpha values of the y position in all three lanes were set to 0.1, and the alpha values were set to 0.6, 0.4, and 0.2 for the x positions in the left, middle, and right lanes, respectively. For APF, because the left and middle lanes slanted to the right, the Rayleigh distribution was used in both the x- and y-directions. In the right lane, the normal distribution was set for the x-direction while the Rayleigh distribution was set for the y-direction. The standard deviations of x and yy were set to (5, 3), (2, 3), and (6, 3) in the left, middle, and right lanes, respectively. The color and contour features are fused into the particle framework [28], so the performance was good. For the proposed RDHOGPF method, the parameters for the motion model in $[x\ y\ \dot{x}\ \dot{y}]^T$ were set to $[10\ 10\ 2\ 2]^T$ and $[6\ 6\ 1\ 1]^T$ in traffic videos 1 and 2, respectively. The total particle number was 100.

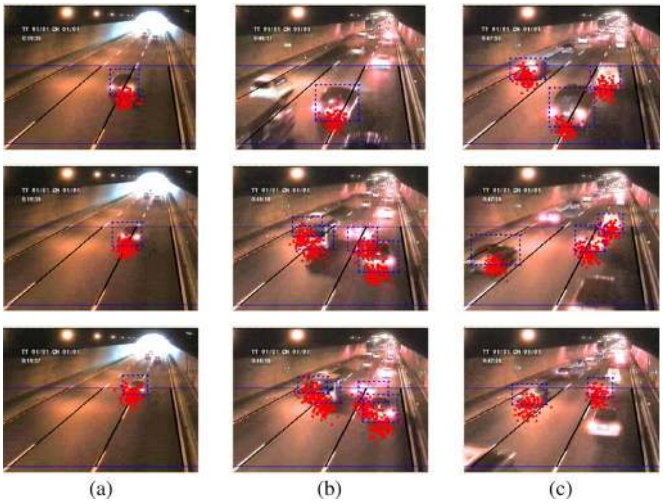


Fig. 8. RDHOGPF tracking results during the lane changing event in traffic video 1. (a) Simple lane changing case. (b) Lane changing with partial occlusion. (c) Lane changing from the middle lane to the right lane with strong rear taillight lighting.

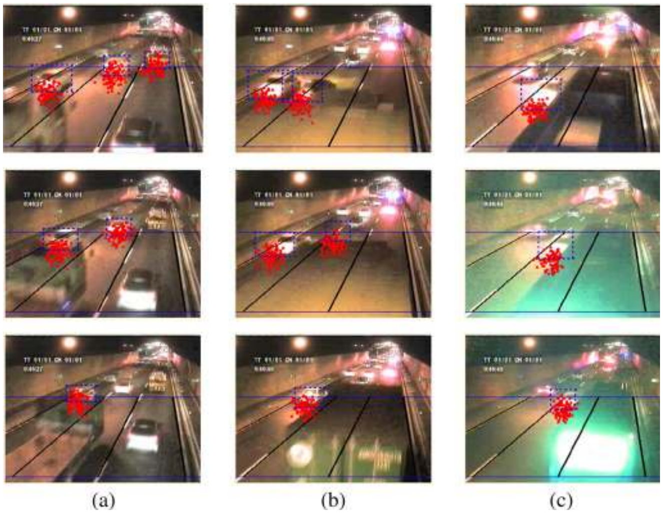


Fig. 9. RDHOGPF tracking results for some complex cases in traffic video 1. (a) Partial occlusion was caused by a trunk in the middle lane. (b) Partial and full occlusions caused by a truck. (c) Occurrence of partial occlusion and auto-white balance at the same time.

Some representative results with the proposed RDHOGPF are shown in Figs. 8 and 9. The RDHOGPF tracking results for the lane changing vehicles are addressed in Fig. 8. The images in Fig. 8(a) illustrate a simple lane change tracking case, while the images in Fig. 8(b) and (c) present that robust tracking was achieved throughout the lane change event, even when partial occlusion occurred.

Fig. 9 are some RDHOGPF tracking results for complex cases. Fig. 9(a) illustrates partial occlusion between the left lane

TABLE IV
DETECTION RESULTS IN TRAFFIC VIDEO 2 (30 min)

Lane	Def.	TSHT [25]	Annealing PF [24]	APF [16]	Contour PF [28]	RDHOGPF
Left Lane	*Tp*	473	444	493	634	638
	Fn	256	285	236	95	91
	Fp	9	117	31	3	1
	Recall	64.88	60.91	67.63	86.97	**87.52**
	Precision	98.13	79.14	94.08	99.53	**99.84**
Middle Lane	*Tp*	664	574	595	676	684
	Fn	55	145	124	43	35
	Fp	25	390	20	6	2
	Recall	92.35	79.83	82.75	94.02	**95.13**
	Precision	96.37	59.54	96.75	99.12	**99.71**
Right Lane	*Tp*	460	422	365	467	482
	Fn	63	101	158	56	41
	Fp	6	314	19	4	2
	Recall	87.95	80.69	69.79	89.29	**92.16**
	Precision	98.71	57.34	95.05	99.15	**99.59**
Total	*Tp*	1597	1440	1453	1777	1804
	Fn	374	531	518	194	167
	Fp	40	821	70	13	5
	Recall	81.02	73.06	73.72	90.16	**91.53**
	Precision	97.56	63.69	95.40	99.27	**99.72**

and middle lane. Fig. 9(b) presents partial and full occlusions caused by a trunk in the right lane, where the tracked vehicles in the middle lane and left lane were counted successfully even when the vehicle in the middle lane was fully covered by the truck. Fig. 9(c) addresses that RDHOGPF handled the partial occlusion successfully and tracked the vehicle even while the auto-white balance function was executed. During the auto-white balance process, the colors in the image changed sharply, and the trajectory updating process became difficult.

Table IV shows the detection results for traffic surveillance video 2, where the test scene was a general road with low contrast because of the brightness of the light. The proposed RDHOGPF could maintain most of the trajectories until the vehicle left the ROI. Some vehicles were not detected, while some trajectories were deleted because they did not contain enough edges, but the overall Recall and Precision for RDHOGPF were both $> 90\%$. In Fig. 10(a)–(c), the images address that the tracked vehicles were tracked successfully when they changed from the left to the middle lane, from the middle to the left lane, and from the middle to the right lane, respectively, even when the vehicles were similar to the background image. Fig. 11 shows some results for occlusion handling during RDHOGPF tracking in traffic video 2. Fig. 11(a) and (b) shows occlusion between the left and middle lanes. The tracked sedan was covered by the roof of the bus in the right lane, but the sedan was counted successfully until the end in Fig. 11(c).

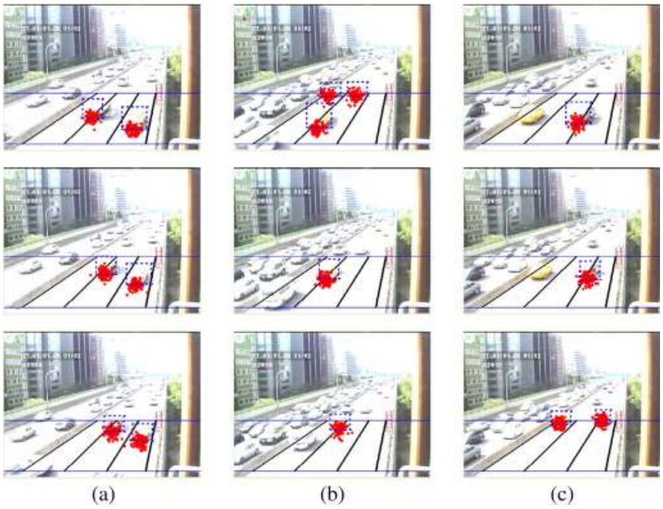


Fig. 10. RDHOGPF tracking results for the lane changing events in traffic video 2. (a) White van changed from the left lane to the middle lane. (b) Taxi changed from the middle lane to the left lane. (c) White sedan with a similar color to the background was still tracked successfully during lane changing.

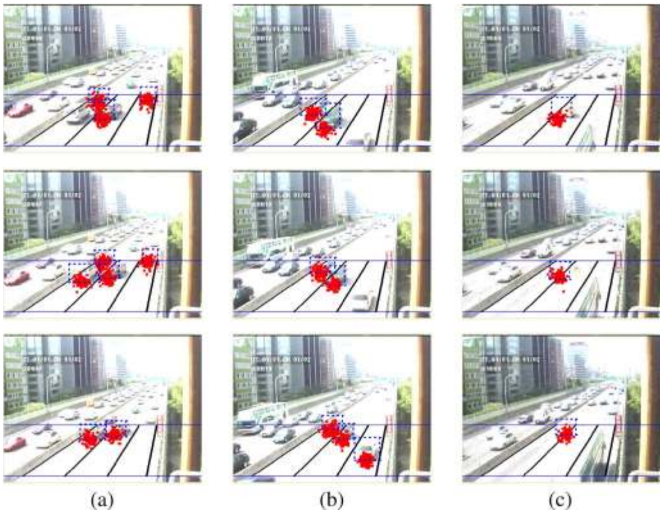


Fig. 11. Occlusion handling results by RDHOGPF in traffic video 2. (a) Partial occlusions between three vehicles. (b) Severe occlusion between two vehicles. (c) Car in the middle lane was covered by the roof of the bus.

Fig. 12 compares the tracking results for traffic video 1 produced using TSHT, annealing PF, APF, and RDHOGPF. The first column in Fig. 12 illustrates that the rectangle size increased with the TSHT method and vehicle detection failed in the middle lane. The particles were more concentrated in annealing PF than in the other methods because of the three-layer enforcement. However, the drift problem still occurred, and the trajectories were deleted when the rectangle contained nothing in the third column. Therefore, three vehicles were missed. In APF, the trajectory in the right lane could not capture all of the vehicle blobs. Therefore, a new trajectory was generated, and this vehicle was eventually counted twice in the third column. Although TSHT tracked a vehicle in the right lane and it was counted correctly, the rectangle size of the tracked vehicle was obviously too large, and the vehicle in the middle lane was missed. In the last column, the vehicles were moving in different lanes, and partial occlusion occurred, but RDHOGPF detected the different vehicles successfully in this complex traffic scene. The comparative videos can be viewed and downloaded at http://140.113.150.201/research/image.php.

The RDHOGPF approach delivered a good performance, but some errors still occurred due to inaccuracies. The truck in Fig. 13(a) was in the middle lane, but it was falsely detected

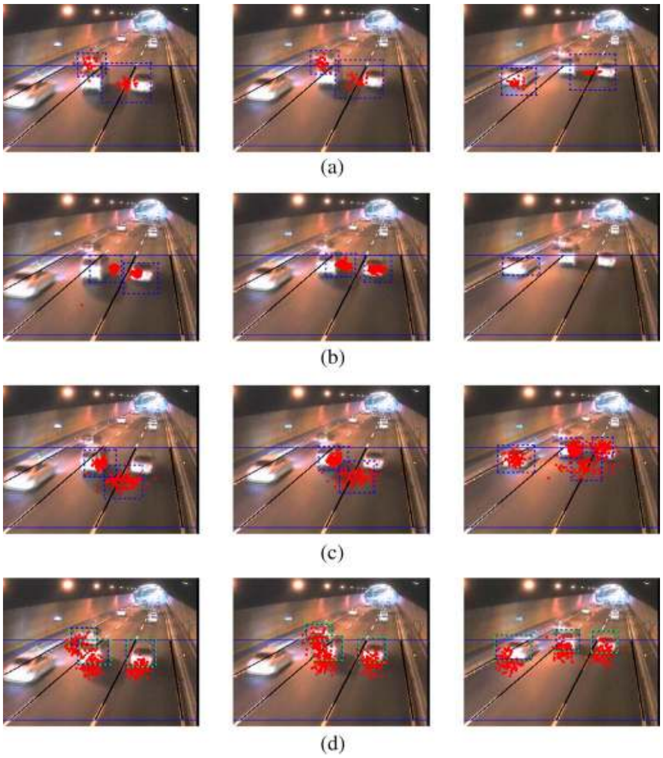


Fig. 12. Comparison of the tracking results using different methods for traffic video 1. (a) TSHT. (b) Annealing PF. (c) APF. (d) RDHOGPF.

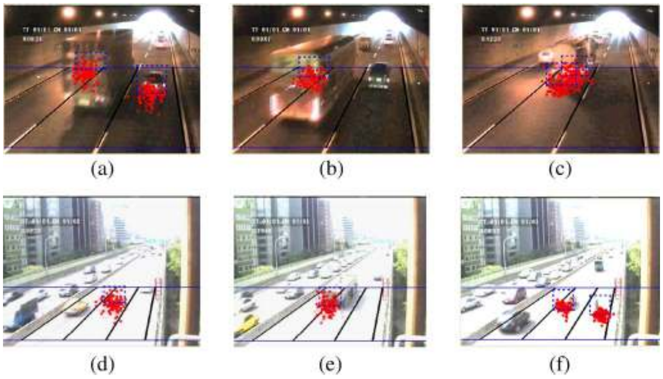


Fig. 13. Failure results with the proposed method. (a) False detection in the left lane. (b) False detection in the left lane and the vehicle was missed in the right lane. (c) Truck was counted twice in the middle lane. (d) White sedan was missed in the right lane. (e) Truck was counted erroneously in the left lane. (f) Motor was falsely detected in the right lane.

and counted in the left lane. In the same situation, the bus was falsely counted in the left lane because the system falsely recognized the bus roof as a vehicle in Fig. 13(b). In Fig. 13(c), the truck was counted twice in the middle lane, and the false positive number increased by one in the middle lane. As shown in Fig. 13(d), some white cars were difficult to recognize when they blended into the road surface. In Fig. 13(e), despite RDHOGPF detecting and tracking the truck successfully, the trajectory was close to the lane mark, so this vehicle was counted in the left lane when it was actually in the middle lane. The car in Fig. 13(f) was falsely detected in the right lane.

To illustrate the results with the proposed RDHOGPF method at crossroads, we provided a test video that involves crossroads. This allowed us to demonstrate the performance of our proposed RDHOGPF method with numerous vehicle direction changes. The test video was derived from the Imagery Library for Intelligent Detection Systems (i-LIDS) [29], which is a well-known data set used to evaluate system performance. Fig. 14 addresses a sequence where a red car changed from a vertical direction to a horizontal direction, and the corresponding appearances of the template and the HOG maps are displayed together. In Fig. 14(a), the red car was tracked, and it changed to the left lane in Fig. 14(b). In Fig. 14(e)–(h), the appearances of the template changed from a front view to a side view, and the corresponding HOG and RDHOG features are

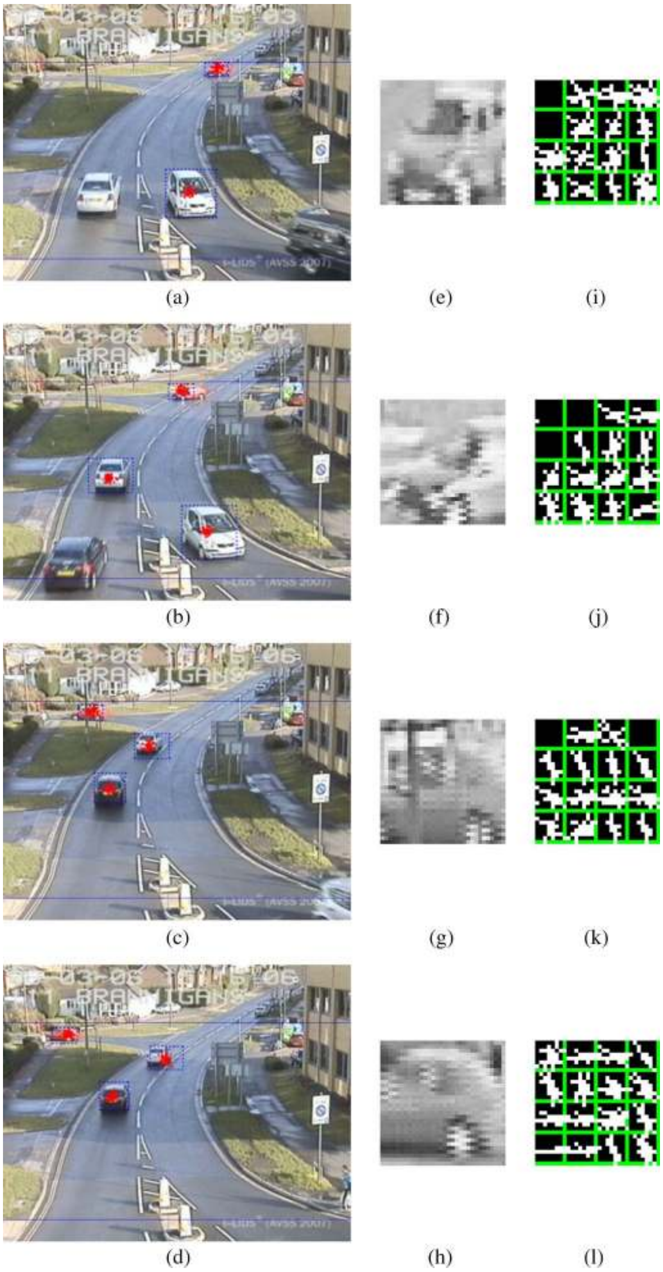


Fig. 14. RDHOGPF tracking results for the red car showing the appearance and RDHOG descriptor updating. (a) Tracking of the red sedan was initiated. (b) Red sedan started to change direction at the crossroads. (c) Red sedan was partially occluded by the utility pole. (d) Presented RDHOGPF method tracked the red sedan until the end of the video. (e)–(h) Appearance of the tracked red sedan. (i)–(l) Corresponding updated RDHOG descriptors.

updated in Fig. 14(i)–(l). In Fig. 14(c), the red car is partially occluded by a utility pole, but the rectangle still located the position of the red car correctly at an appropriate size. The lane on the left was oriented in a horizontal direction, so the scaling factor is set to one. Thus, the size of the rectangle was not changed in this lane. Finally, in Fig. 14(d), when the red car passed the utility pole, the proposed RDHOGPF method was not affected by the utility pole, and it tracked the red car successfully.

V. Conclusion

This paper proposed an effective vehicle detection and tracking approach based on RDHOG with occlusion handling. Background subtraction was used to detect vehicles by segmenting the foreground image and verifying their edge features. The trajectory was generated if the vehicle was detected three times in consecutive images. The target pattern was normalized to produce a 32×32 image before the HOG and RDHOG features were extracted. RDHOG was based on the HOG descriptor, which was used to represent the difference between the central block and the surrounding blocks of the target. After combining the features of HOG and RDHOG, the descriptor could describe the shape of the target and discriminate between the center and borders. To track the vehicle with a correct rectangle size, each particle had two sample images. One was of the original size, whereas the other one had a scaled width and height. The position of the tracked vehicle was estimated and updated after calculating the likelihoods for vehicle tracking. We also developed a method for occlusion handling to reduce the effects of occluded vehicles by increasing the difference values for the partially obscured region. The rectangle of an older trajectory was masked, and smaller particle weights were obtained when a particle occupied this masked occluded region, so the trajectories were prevented from moving unexpectedly. Using RDHOG tracking, the drift effects were reduced in scenes with low viewing angles, low contrast, and excessive brightness. Despite changes in the appearance of vehicles when negotiating crossroads, the proposed RDHOGPF method could track the vehicle correctly. The RDHOGPF method could also track vehicles accurately when the tracked vehicles were occluded by signposts and utility poles. The RDHOGPF vehicle tracker was implemented in C++ and tested using real highway traffic scenes captured inside a tunnel, on a general road with low contrast, and an i-LIDS image obtained from the U.K. government. Our experimental results and comparisons with other tracking methods demonstrated that RDHOGPF is highly effective and it delivers good performance in many different traffic situations.

Acknowledgment

The authors would like to thank the laboratory members who helped to manually count the vehicles in the video sequences.

References

[1] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[2] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011.

[3] Y. L. Chen, B. F. Wu, H. Y. Huang, and C. J. Fan, "A real-time vision system for nighttime vehicle detection and traffic surveillance," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 2030–2044, May 2011.

[4] B. F. Wu and J. H. Juang, "Adaptive vehicle detector approach for complex environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 817–827, Jun. 2012.

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

[6] S. Chen, J. Zhang, Y. Li, and J. Zhang, "A hierarchical model incorporating segmented regions and pixel descriptors for video background subtraction," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 118–127, Feb. 2012.

[7] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, Jul. 2008.

[8] 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.

[9] W. Kim and C. Kim, "Background subtraction for dynamic texture scenes using fuzzy color histograms," *IEEE Trans. Signal Process. Lett.*, vol. 19, no. 3, pp. 127–130, Mar. 2012.

[10] B. Han and L. S. Davis, "Density-based multifeature background subtraction with support vector machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1017–1023, May 2012.

[11] S. Y. Chen, "Kalman filter for robot vision: A survey," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4409–4420, Nov. 2012.

[12] J. Garcia, A. Gardel, I. Barvo, J. L. Lazaro, M. Martinez, and D. Rodriguez, "Directional people counter based on head tracking," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3991–4000, Sep. 2013.

[13] F. Lin, X. Dong, B. M. Chen, K.-Y. Lum, and T. H. Lee, "A robust real-time embedded vision system on an unmanned rotorcraft for ground target following," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1038–1049, Feb. 2012.

[14] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[15] Y. Lao, J. Zhu, and Y. F. Zheng, "Sequential particle generation for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, Sep. 2009.

[16] J. Scharcanski, A. B. de Oliveria, P. G. Cavalcanti, and Y. Yari, "A particle-filtering approach for vehicular tracking adaptive to occlusions," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 381–389, Feb. 2011.

[17] N. Bouaynaya and D. Schonfeld, "On the optimality of motion-based particle filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1068–1072, Jul. 2009.

[18] P. Pan and D. Schonfeld, "Visual tracking using high-order particle filtering," *IEEE Trans. Signal Process. Lett.*, vol. 18, no. 1, pp. 51–54, Jan. 2011.

[19] P. L. M. Boutterfroy, A. Bouzerdoum, S. L. Phung, and A. Beghdadi, "Integrating the projective transform with particle filtering for visual tracking," *EURASIP J. Image Video Process.*, vol. 2011, pp. 1–11, Jan. 2011.

[20] Z. H. Khan, I. Y. H. Gu, and A. G. Backhouse, "Robust visual tracking using multi-mode anisotropic mean shift and particle filters," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 1, pp. 74–87, Jan. 2011.

[21] X. Li, T. Zhang, X. Shen, and J. Sun, "Object tracking using an adaptive Kalman filter combined with mean shift," *Opt. Eng.*, vol. 49, no. 2, pp. 1–3, 2010.

[22] A. Makris, D. Kosmopoulos, S. Perantonis, and S. Theodoridis, "A hierarchical feature fusion framework for adaptive visual tracking," *Image and Vision Computing*, vol. 29, no. 9, pp. 594–606, 2011.

[23] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, Nov. 2011.

[24] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filter," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2000, vol. 2, pp. 126–133.

[25] E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, vol. 2, pp. 221–224.

[26] D. Monzo, A. Albiol, A. Albiol, and J. M. Mossi, "Color HOG-EBGM face recognition," in *Proc. IEEE Int. Conf. Image Process.*, 2011, pp. 785–788.

[27] J. Arrospide, L. Salgado, and J. Marinas, "HOG-like gradient-based descriptor for visual vehicle detection," in *Proc. IEEE Intell. Veh. Symp.*, 2012, pp. 223–228.

[28] J. Li, X. Lu, L. Ding, and H. Lu, "Moving target tracking via particle filter based on color and contour features," in *Proc. 2nd Int. Conf. Inf. Eng. Comput. Sci.*, 2012, pp. 1–4.

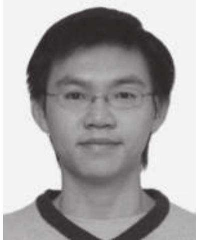
[29] [Online]. Available: https://www.gov.uk/imagery-library-for-intelligent-detection-systems#i-lids-datasets

[30] C.-Y. Chang and H. W. Lie, "Real-time visual tracking and measurement to control fast dynamics of overhead cranes," *IEEE Trans. Ind. Electron.*, vol. 59, no. 3, pp. 1640–1649, Mar. 2012.

Bing-Fei Wu (S'89–M'92–SM'02–F'12) received the B.S. and M.S. degrees in control engineering from 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, CA, USA, in 1992.

He was with the Department of Electrical Engineering, NCTU, in 1992, was promoted to Professor in 1998, and Distinguished Professor in 2010. He has served as the Director of the Institute of Electrical and Control Engineering, NCTU, since 2011. His current research interests include image recognition, vehicle driving safety, intelligent control, intelligent transportation systems, and embedded systems.

Dr. Wu founded and served as the Chair of the IEEE Systems, Man, and Cybernetics Society's (SMC) Taipei Chapter in 2003 and was elected as the Chair of the Technical Committee on Intelligent Transportation Systems of the IEEE SMC Society in 2011.



Chih-Chung Kao (S'06) received the B.S. degree from the Department of Electrical Engineering, National Taiwan Ocean University, Keelung, Taiwan, in 2004 and the M.S. degree from the Institute of Industrial Education, National Taiwan Normal University, Taipei, Taiwan, in 2006. He is currently working toward the Ph.D. degree at National Chiao Tung University, Hsinchu, Taiwan.

His research interest includes image processing, pattern recognition, statistical learning, and intelligent transportation systems.

Cheng-Lung Jen (S'06) received the B.S. degree in electrical engineering from the National Chin-Yi University of Technology, Taichung, Taiwan, in 2004 and the M.S. degree in electrical engineering from the National Central University, Jhongli, Taiwan, in 2006. He is currently working toward the Ph.D. degree at National Chiao Tung University, Hsinchu, Taiwan.

His main research interests lie in the area of machine learning, wireless positioning systems, and robotics.

Yen-Feng Li was born in Taitung, Taiwan, in 1976. He received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2007 and the M.S. degree in electronic engineering from National Chin-Yi University of Technology, Taichung, Taiwan, in 2009. He is currently working toward the Ph.D. degree in electrical engineering at National Chiao Tung University, Hsinchu, Taiwan.

His research interests include image processing, embedded systems, and intelligent transportation system.

Ying-Han Chen was born in Tainan, Taiwan, in 1981. He received the B.S. and M.S. degrees in electrical engineering from National Central University, Jhongli, Taiwan, in 2003 and 2006, respectively. He is currently working toward the Ph.D. degree in electrical engineering at National Chiao Tung University, Hsinchu, Taiwan.

His research interests include networking, embedded systems, and digital signal processing.

Jhy-Hong Juang was born in I-Lan, Taiwan, in 1974. He received the B.S. degree in control engineering, and the M.S. and Ph.D. degrees in electrical and control engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1997, 1999, and 2012, respectively.

He is currently with NCTU. His research interests include image processing, software engineering, and pattern recognition.