Model-Based Vehicle Localization Based on 3-D Constrained Multiple-Kernel Tracking

Kuan-Hui Lee, Jenq-Neng Hwang, *Fellow, IEEE*, and Shih-I Chen

***Abstract*— In this paper, we propose a novel model-based vehicle localization approach on the basis of surveillance cameras. The proposed approach regards each patch of the 3-D vehicle model as a kernel, and tracks the kernels under certain constrains facilitated by the 3-D geometry of the vehicle model. Meanwhile, a kernel density estimator is designed to well fit the 3-D vehicle model during tracking. With elegant application of the constrained multiple-kernel tracking facilitated with the 3-D vehicle model, the vehicles are able to be tracked efficiently and located precisely. The proposed approach achieves high effectiveness in the tracking and localization by taking advantage of the color similarity and shape fitness. Experimental results have shown the favorable performance of the proposed approach, in several scenarios, which efficiently tracks vehicles while maintaining the knowledge of 3-D geometry of the tracked vehicles.**

***Index Terms*— 3-D vehicle modeling, kernel-based tracking, vehicle localization, vehicle tracking, visual surveillance.**

I. Introduction

THE development of traffic surveillance plays an important role in intelligent transportation systems, and video-based traffic surveillance over vehicles has thus become a very important research area. By tracking vehicles, it is possible to collect their moving trajectories in videos for high-level analytics, such as, detection and avoidance of vehicle accidents, detection of specific vehicles for theft discovery, detection of vehicle with unusual behavior, collection of traffic statistics information for further decision making, and so on. Therefore, researchers are motivated to develop effective vehicle tracking systems, which are not only robust and fast but also able to collect the information efficiently and accurately.

Vehicle tracking approaches using surveillance cameras are roughly divided into two categories [1]: 2-D based and 3-D based approaches. In 2-D based approaches, 2-D motions and trajectories are commonly used in the Kalman-filtering or particle-filtering frameworks, mainly according to 2-D geometry features, such as edges, lines, and contours [2]–[6]. Hsieh *et al.* [7] use area, length and a set of rules, and a linearity feature to roughly measure the vehicle silhouette. Leibe *et al.* [8] propose a method to detect 2-D points with associated descriptors of local appearance, which are matched with those of vehicles from a previously learned lookup table. Several literatures combine corners or interest points with feature descriptors, such as SIFT [9] and HOGs [10], [11] for the tracking. Alternatively, some approaches [12]–[14] include color histogram as tracking features, since it is not only more invariant to vehicle rotations and translations but also more robust to occlusions and noise.

The 2-D-based vehicle tracking can be regarded as a special case of video object tracking, which has been extensively developed and investigated. A framework called kernel-based object tracking proposed by Comaniciu *et al.* [15], has attracted great attention in the past few years owing to its using, with relatively low computation, the spatially weighted color histogram-based features. The basic concept is to maximize the similarity between the target's and the candidate's appearance models (i.e., color histogram), which are built by spatially masking the object with a kernel function. With this framework, many literatures have been proposed to improve the tracking performance [16]–[19]. Furthermore, the multiple-kernel tracking, extended from single-kernel approaches and connecting the multiple collaborative kernels by using certain predefined constraints, has been also proposed [20], [21]. Chu *et al.* [22] generalize the constrained multiple-kernel (CMK) tracking by adaptively adjusting kernels' weights according to their similarity, to improve the reliability when the occlusion occurs.

Nevertheless, the challenge of vehicle tracking lies in the fact that color or texture information is not very discriminative among different vehicles, so that these object-tracking techniques may not be directly applicable for vehicle tracking. In addition, only color information is insufficient to locate the vehicles accurately. An intuitive way to extract features for vehicle identity is to modularize a vehicle into a 3-D model, resulting in the development of the 3-D based approaches, which can be roughly categorized into two types [23]: feature-based and intensity-based. The feature-based 3-D approaches use 2-D geometry features to evaluate the constructed 3-D models. The approaches proposed in [24] and [25] track vehicles in an extended Kalman-filtering framework with a 3-D model built by edges and corresponding features. In [26], the approach evaluates the distance between extracted edge points and model projection. Liebelt *et al.* [27] extend the idea used in [8] to detect the 3-D poses of vehicles from 2-D contours. The approaches which evaluate the similarity of the projected contours have also been proposed in [28] and [29]. Because 2-D feature extraction is sensitive to image noise and occlusion, the intensity-based approaches evaluate image

Manuscript received December 5, 2013; revised March 4, 2014; accepted May 8, 2014. Date of publication June 5, 2014; date of current version January 5, 2015. This paper was recommended by Associate Editor W. Zhang.

K.-H. Lee is with the Information Processing Laboratory, Department of Electrical Engineering, University of Washington, Seattle, WA 98105 USA (e-mail: ykhlee@uw.edu).

J.-N. Hwang is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98105 USA (e-mail: hwang@uw.edu).

S.-I. Chen is with the Institute for Information Industry, Taipei 10622, Taiwan (e-mail: shihi@iii.org.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/TCSVT.2014.2329355

intensities or gradients by projecting the 3-D models onto the images. Brisdon firstly proposes the ICONIC method [30] to evaluate the difference of gray levels at uniformly sampled pixels in a statistical framework. Ferryman *et al.* [31] present a deformable model with 29 parameters, combined with a principal component analysis framework, to perform the vehicle tracking. Tan *et al.* [32], [33] estimate orientation of a vehicle by gradient vectors and evaluate the poses by intensity values. Zhang *et al.* [34] propose a deformable-model based tracking approach to dynamically build 3-D vehicle models and locate vehicles in an iterative framework. Zheng and Peng [23] propose an efficient vehicle pose estimation scheme, to reduce the search range of the tracked poses. The approach in [35] develops a vehicle-tracking system on the basis of a highly complicated 3-D deformable model. In [36], a 3-D deformable vehicle model based on the model fitting with weighted Jacobian is effectively proposed for an image-based vehicle retrieval system. However, these intensity-based approaches focus solely on the shape fitting, without considering the color information, to perform the tracking. In addition, most of the proposed pose estimation schemes are time consuming owing to the optimization of the parameters.

Taking advantage of kernel-based tracking and 3-D vehicle modeling of the tracked vehicle, our previous work [37] incorporates 3-D vehicle models into the multiple-kernel-based tracking. The system adopts 2-D CMK tracking framework [22], facilitated with an automatically built 3-D vehicle model, to efficiently track and localize the vehicles. Even though the approach in [37] performs well in limited scenarios, there are some issues need to be improved.

1) As the scale or the orientation of vehicles changes greatly, the information of the kernels becomes unreliable and has to be updated.
2) The tracking of the kernels is based on the 2-D (image) space, because the color information is extracted from the 2-D image frame, while moving of the 3-D vehicle model is still defined in the 3-D space.
3) The constraints for multiple kernels in 2-D space are not suitable to the kernels in 3-D space, since they will change during the movement of a vehicle owing to the varying view aspects.

These issues may gradually make unreliable the tracking features of the vehicles, so that the error of the tracking becomes larger and larger. To overcome this problem, in this paper, we propose an innovative 3-D constrained multiple-kernel tracking facilitated with the 3-D vehicle model. Instead of tracking kernels in the 2-D space, the proposed approach directly tracks the vehicles in 3-D space. Owing to the 3-D constraints, multiple kernels are tied conditionally and mean-shifted properly according to the color information back-projected from the image. Meanwhile, a kernel density estimator is designed for accurately fitting the shape of the projected 3-D vehicle model. Therefore, the proposed approach not only tracks the vehicles on the basis of the color information, but also locates the 3-D vehicle models according to the shape fitness evaluation.

The rest of this paper is organized as follows. In Section II, the adopted algorithms including the CMK tracking and the 3-D vehicle modeling are reviewed. Section III depicts how to integrate the 3-D vehicle model into the CMK tracking framework, including 2-D constrained and 3-D constrained schemes. In Section IV, we describe the complete system of the proposed 3-D CMK tracking for vehicles. The experimental results are shown in Section V, followed by the conclusion in Section VI.

II. Review of Adopted Algorithms

In this section, we briefly describe two algorithms adopted in the proposed work: the CMK tracking framework proposed in [22], and the 3-D vehicle modeling proposed in [34].

A. CMK Tracking

The objective of the CMK tracking is to retrieve a candidate object, which can be described as multiple kernels with prespecified constraints among these kernels, so that the maximum similarity between the tracked object and the candidate model can be reached. For N_k kernels, the total cost function $J(\mathbf{x})$ is defined to be the sum of the N_k individual cost functions $J_\kappa(\mathbf{x})$, which is designed to be inversely proportional to the similarity

$$J(\mathbf{x}) = \sum_{\kappa=1}^{N_k} J_\kappa(\mathbf{x}),\ J_\kappa(\mathbf{x}) \propto 1/\mathrm{simi}_\kappa(\mathbf{x}) \tag{1}$$

where $\mathrm{simi}_\kappa(\mathbf{x})$ is the similarity function at the location $\mathbf{x}$ in the state space domain.

Moreover, the constraint functions $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ need to be considered to maintain the relative locations of the kernels. The constraint functions confine the kernels based on their spatial interrelationships. Thus, the problem could be further formulated by

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} J(\mathbf{x}), \quad \text{subject to } \mathbf{C}(\mathbf{X}) = 0. \tag{2}$$

To gradually decrease the total cost function and maintain the constraints satisfied during the state search, the movement vector $\delta_{\mathbf{x}}$, that is, the gradient vector of the $J(\mathbf{x})$, is needed for using the projected gradient method [22] to iteratively solve the constrained optimization problem. The basic idea is to project the gradient vector onto two orthogonal spaces, one is related to decreasing the cost function and the other corresponds to satisfying the constraints. (i.e., $\mathbf{C}(\mathbf{x}) = \mathbf{0}$)

$$\begin{aligned}\delta_{\mathbf{x}} &= \alpha\big(-\mathbf{I}+\mathbf{C}_{\mathbf{x}}\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\mathbf{C}_{\mathbf{x}}^T\big)\mathbf{J}_{\mathbf{x}} + \big(-\mathbf{C}_{\mathbf{x}}\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\mathbf{C}(\mathbf{x})\big)\\ &= \delta_{\mathbf{x}}^A + \delta_{\mathbf{x}}^B\end{aligned} \tag{3}$$

where $\mathbf{x} \in \mathrm{R}^n$ is the state vector, $\mathbf{C}(\mathbf{x}) = [c_1(\mathbf{x}) \cdots c_m(\mathbf{x})]^T$ is the matrix including m constraint functions, and $c_i(\mathbf{x}) : \mathrm{R}^n \rightarrow \mathrm{R}$ is the ith constraint function, $\mathbf{C}_{\mathbf{x}} \in \mathrm{R}^{n\times m}$ is the gradient matrix of constraint functions with respect to $\mathbf{x}$, $\mathbf{J}_{\mathbf{x}}$ is the gradient vector of the total cost function with respect to $\mathbf{x}$, and $\alpha > 0$ is the step size.

As proved in [22], $\delta_{\mathbf{x}}^A$ and $\delta_{\mathbf{x}}^B$ are orthogonal to each other. Moving along the $\delta_{\mathbf{x}}^A$ decreases the total cost function $J(\mathbf{x})$ while keeping the same values of $\mathbf{C}(\mathbf{x})$. On the other hand, moving along the $\delta_{\mathbf{x}}^B$ can lower the absolute values of $\mathbf{C}(\mathbf{x})$. Owing to these characteristics, the optimal solution can be

reached in an iterative manner. The iteration is stopped until either the cost function and the absolute values of constraint functions are both lower than some given thresholds ε_j and ε_c, respectively, or the iteration count is larger than a threshold T [22, Algorithm 1].

When occlusion occurs, not all the kernels can be used for matching. To solve the issue, each kernel is assigned with an adaptively adjustable weight w_κ in (1)

$$J(\mathbf{x}) = \sum_{\kappa=1}^{N_k} w_\kappa \cdot J_\kappa(\mathbf{x}). \tag{4}$$

Thus, the movement vector in (3) is modified to be

$$\begin{aligned}\delta_{\mathbf{x}} = \alpha\big(-\mathbf{I} + \mathbf{C}_{\mathbf{x}}\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\mathbf{C}_{\mathbf{x}}^T\big)\mathbf{W}\mathbf{J}_{\mathbf{x}} \\ +\big(-\mathbf{C}_{\mathbf{x}}\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\mathbf{C}(\mathbf{x})\big)\end{aligned} \tag{5}$$

where

$$\mathbf{W} = \begin{bmatrix} w_1\mathbf{I} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_{N_k}\mathbf{I} \end{bmatrix}$$

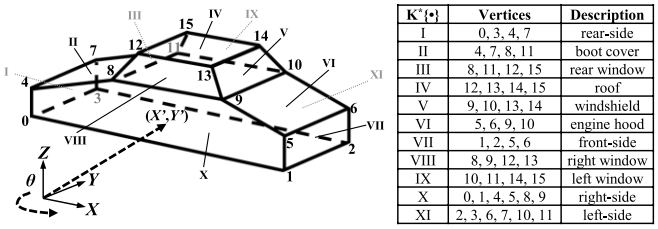
and $w_\kappa \propto \text{simi}_\kappa(\mathbf{x})$, $\mathbf{I}$ is an $n/N_k \times n/N_k$ identity matrix, and n is the dimension of the state space.

The value w_κ corresponding to the κth kernel is adaptively updated on the basis of the similarity simi_κ and is normalized to make the sum equal to N_κ. The idea of the adaptation is that the movement vector will have more confidence on a kernel with higher similarity than that with the lower one. When a kernel is ineffective because of occlusion, the movement vector will adaptively count on the other effective kernels (i.e., kernels with higher similarity) [22].

B. 3-D Vehicle Modeling

3-D vehicle modeling [34] is to generate an approximate 3-D vehicle model deformed from a 3-D generic model. The basic assumption is that the camera is static and well calibrated (i.e., the 3×4 projective matrix $\hat{P}$ is known), to facilitate the projection from the 3-D model to 2-D geometric primitives. The 3-D deformable model with 16 vertices and 23 arcs, as shown in Fig. 1(a) [34], is defined by 12 shape parameters, such as vehicle length (L), vehicle widths (W1, W2), vehicle heights (H1, H2, H3, H4), and so on; its pose is determined by three parameters, which are its position (X', Y') on the ground plane, and its orientation θ about the vertical axis perpendicular to ground-plane, based on the ground-plane constraints (GPC) [32]. These 15 parameters in total can be estimated by evaluating the fitness between image data and the projection of a 3-D deformable model, in an evolutionary computing framework called estimation of multivariate normal algorithm-global ($\text{EMNA}_{\text{global}}$) [38].

A fitness evaluation score (FES) is considered as the measurement of the 3-D vehicle model fitting in an iterative optimization. To evaluate FES, the wire-frame 3-D vehicle model is projected onto the image plane to construct a series of visible projected line segments. For every visible projected line segment l, a virtual rectangle with size of $L_r \times 2W_r$ is assigned, as shown in Fig. 2. The idea of the fitness evaluation



K*{•}	Vertices	Description
I	0, 3, 4, 7	rear-side
II	4, 7, 8, 11	boot cover
III	8, 11, 12, 15	rear window
IV	12, 13, 14, 15	roof
V	9, 10, 13, 14	windshield
VI	5, 6, 9, 10	engine hood
VII	1, 2, 5, 6	front-side
VIII	8, 9, 12, 13	right window
IX	10, 11, 14, 15	left window
X	0, 1, 4, 5, 8, 9	right-side
XI	2, 3, 6, 7, 10, 11	left-side

(a) (b)

Fig. 1. (a) Generic model for 3-D vehicle modeling [34]. (b) Table of the kernels in 3-D vehicle model.

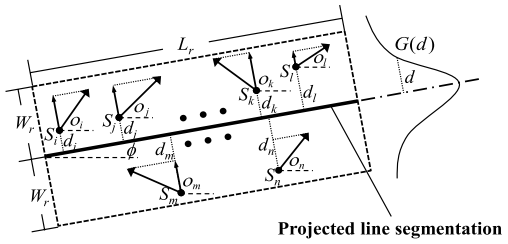


Fig. 2. Idea of fitness evaluation, adopted from [34, Fig. 4(a)].

is to compare gradients along the directions perpendicular to the projected line segments. If a line segment fits the image data well, the gradient directions of pixels with large gradient magnitude values in the rectangle should concentrate on the perpendicular direction of this projected line. The gradient magnitude $m(x, y)$ and orientation $o(x, y)$ are calculated for each pixel S_i within the virtual rectangle. More specifically, the fitness score E_{S_i} for S_i is calculated by the vertical component of its gradient magnitude along the line segment direction with by angel ϕ

$$E_{S_i} = |m(x, y) \cdot \sin(o(x, y) - \phi)|. \tag{6}$$

Note that E_{S_i} is further weighted by a standard Gaussian distribution $G_{0,w}(d_i)$ with $\mu = 0$ and $\sigma = w$, where d_i denotes the distance between S_i and the projected line segment. Hence, the fitness score of the project line segment l is calculated by the weighted sum E_{S_i}, and the whole FES between the projection of the vehicle model and the image data is accumulated by all the visible lines

$$E = \sum_l [\log(E_l)] = \sum_l \log\left(\sum_{S_i} [E_{S_i} \cdot G_{0,w}(d_i)]\right). \tag{7}$$

III. CMK Tracking With 3-D Vehicle Modeling

To combine the CMK tracking with 3-D vehicle modeling, we regard each patch (plane) of the 3-D model as a kernel. The corresponding vertices of each kernel, annotated by $\text{K}^*\{\bullet\}$ (superscript indicates the dimension of the kernel space), are shown in Fig. 3. Each kernel is a basic component in the tracking procedure.

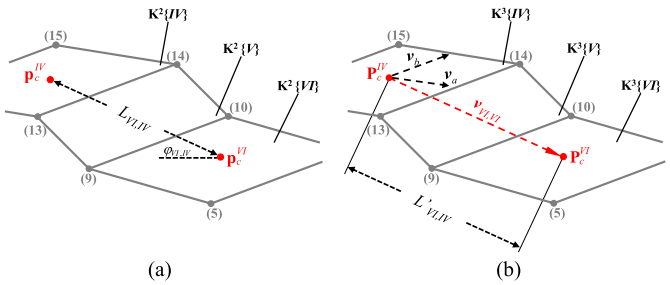


Fig. 3. Example of the constraints between K*{*VI*} and K*{*IV*} in (a) 2-D and (b) 3-D cases.

A. Multiple-Kernel in 2-D Space

After a 3-D vehicle model is built, it is projected onto the image to form a 2-D wire-frame vehicle model, which includes many kernels ($\mathrm{K}^2\{\cdot\}$). With the framework in [15], the color histograms of the kernels are used in the measurement of the similarity. Hence, $\mathrm{K}^2\{\kappa\}$ (also called the target kernel) is described by its probability density function (pdf) q in terms of r-bin histograms

$$q_u^\kappa = \frac{\sum_{i=1}^{n_\kappa} k\left(\left\|\frac{\mathbf{p}^\kappa - \mathbf{p}_i^\kappa}{h}\right\|^2\right)\delta\left[b(\mathbf{p}_i^\kappa) - u\right]}{\sum_{i=1}^{n_\kappa} k\left(\left\|\frac{\mathbf{p}^\kappa - \mathbf{p}_i^\kappa}{h}\right\|^2\right)}, \sum_{u=1}^{r} q_u^\kappa = 1 \quad (8)$$

where $|\cdot|$ is L2 norm, $\mathbf{p}^\kappa \in \mathrm{R}^2$ is the center of $\mathrm{K}^2\{\kappa\}$, $\{\mathbf{p}_i^\kappa\}_{i=1\ldots n_\kappa}$, $\mathbf{p}_i^\kappa \in \mathrm{R}^2$ are the pixel locations inside $\mathrm{K}^2\{\kappa\}$, h is the bandwidth of the $\mathrm{K}^2\{\kappa\}$, and δ is the Kronecker delta function. The function $b : \mathrm{R}^2 \rightarrow \{1 \cdot r\}$ associates the pixel at location $\mathbf{p}_i$ with the index $b(\mathbf{p}_i^\kappa)$ of its bin in the color histograms. $k(\cdot)$ is a kernel function with a convex and monotonic decreasing kernel profile.

During multiple-kernel tracking, all the kernels search the regions which have most similarity to the target kernel, while maintaining the given constraints. However, either due to the view aspects or occlusions, not all the kernels are reliable. First, according to the view aspect, only completely visible planes projected onto the image frame are used for multiple kernel tracking. In other words, kernels hidden behind are weighted by zero. Second, to keep the interrelationship of the kernels, several constraints need to be assigned according to the 2-D geometry between the kernels. Hence, the constraint functions are assigned by

$$\begin{cases} \left\|\mathbf{p}_c^\kappa - \mathbf{p}_c^{IV}\right\|^2 = (L_{\kappa,IV})^2 \\ \frac{y^\kappa - y^{IV}}{x^\kappa - x^{IV}} = \tan^{-1}(\varphi_{\kappa,IV}) \end{cases} \text{for any visible } K^2\{\kappa|\kappa \neq IV\} \quad (9)$$

where $\mathbf{p}^{IV} = (x^{IV}, y^{IV})$ is the center of $\mathrm{K}^2\{IV\}$ and $\{(x^\kappa, y^\kappa)\}$ are the centers of the visible kernels $\mathrm{K}^2\{\kappa\}$, $L_{\kappa,IV}$ is the initial distance between the centers of $\mathrm{K}^2\{\kappa\}$ and $\mathrm{K}^2\{IV\}$, and $\varphi_{\kappa,IV}$ is the initial angle between the kernel axis and the horizontal axis. These constants need to be adaptively updated when size and/or orientation of a vehicle are/is greater than the empirical thresholds. Fig. 3(a) shows an example of $\kappa = VI$, where $(\cdots)$ is the index of the vertices shown in Fig. 1(a). Notice that both $L_{VI,IV}$ and $\varphi_{VI,IV}$ are measured in 2-D space.

Although this method performs well in limited scenarios, it still suffers from performance degradations (due to the three issues discussed in Section I) in more general scenarios. To overcome these issues, we propose a new approach, which extends the kernel descriptions from 2-D spaces to 3-D space. Besides, a kernel density estimator is designed to improve the fitness of the 3-D vehicle model during tracking. This is regarded as a projected gradient of the cost function in (3).

B. Multiple-Kernel in 3-D Space

The basic concept is to consider 3-D planes as kernels and search their candidate kernels in 3-D space instead of 2-D space. In other words, we minimize the cost function $J(\mathbf{x})$ in (4), where $\mathbf{x} \in \mathrm{R}^3$. To make it clearer, we redefine the formula and the annotations for this problem. Let $\mathbf{p}$ be the 2-D points in the images, and $\mathbf{P}$ be the corresponding 3-D points, where $\mathbf{p} \in \mathrm{R}^2$, $\mathbf{P} \in \mathrm{R}^3$, $\mathbf{p}$ is obtained by projecting $\mathbf{P}$ through the projective matrix $\hat{P}$.

To describe a kernel in 3-D space, we need to provide color information to a 3-D kernel ($\mathrm{K}^3\{\cdots\}$), because the color information from the images is only defined in 2-D space. Thus, for each visible kernel $\mathrm{K}^3\{\kappa\}$, we associate the color information by back-projecting the 2-D points within $\mathrm{K}^2\{\kappa\}$ to the 3-D points within $\mathrm{K}^3\{\kappa\}$. Let $\vec{\mathbf{P}}^\kappa$ denote the 3-D points back-projected from 2-D point $\mathbf{p}^\kappa$ in the image to the 3-D kernel (or plane) $\mathrm{K}^3\{\kappa\}$, that is, the intersection between a viewing ray passing through $\mathbf{p}^\kappa$ and the plane $\mathrm{K}^3\{\kappa\}$. Therefore, $\mathrm{K}^3\{\kappa\}$ can now be described by its pdf q in terms of the r-bin histograms

$$q_u^\kappa = \frac{\sum_{i=1}^{n_\kappa} k\left(\left\|\frac{\mathbf{P}_c^\kappa - \overleftarrow{\mathbf{P}}_i^\kappa}{h'}\right\|^2\right)\delta\left[b(\mathbf{p}_i^\kappa) - u\right]}{\sum_{i=1}^{n_\kappa} k\left(\left\|\frac{\mathbf{P}_c^\kappa - \overleftarrow{\mathbf{P}}_i^\kappa}{h'}\right\|^2\right)}, \quad \sum_{u=1}^{r} q_u^\kappa = 1 \quad (10)$$

where $\{\overleftarrow{\mathbf{P}}_i^\kappa\}_{i=1\ldots n_\kappa}$, $\overleftarrow{\mathbf{P}}_i^\kappa \in \mathrm{R}^3$ are the 3-D points back-projected from $\{\mathbf{p}_i^\kappa\}$ (inside $\mathrm{K}^2\{\kappa\}$) to $\mathrm{K}^3\{\kappa\}$, $\mathbf{P}_c^\kappa$ is the center and h' is the bandwidth of $\mathrm{K}^3\{\kappa\}$.

A reference kernel (annotated by $\mathrm{K}^3\{\kappa^*\}$) which has the maximum visible area is selected within $\mathrm{K}^3\{I, II, \ldots, IX\}$. To properly set the constraints, we consider the 3-D geometry of the vehicle model. First, the distance between $\mathrm{K}^3\{\kappa\}$ and $\mathrm{K}^3\{\kappa^*\}$ should be the same, which implies

$$\left\|\mathbf{P}_c^\kappa - \mathbf{P}_c^{\kappa^*}\right\|^2 = (L'_{\kappa,\kappa^*})^2, \text{for any visible } \mathrm{K}^3\{\kappa|\kappa \neq \kappa^*\} \quad (11)$$

where L'_{κ,κ^*} is the initial distance between $\mathrm{K}^3\{\kappa\}$ and $\mathrm{K}^3\{\kappa^*\}$. Second, the pitch and yaw between $\mathrm{K}^3\{\kappa^*\}$ and $\mathrm{K}^3\{\kappa\}$ should be the same. We start to calculate the two vectors v_a and v_b which are orthogonal to each other and cross $\mathbf{P}_c^{\kappa^*}$

$$v_a = \frac{\mathbf{P}_a^\kappa + \mathbf{P}_o^\kappa}{2} - \mathbf{P}_c^{\kappa^*}, \; v_b = \frac{\mathbf{P}_b^\kappa + \mathbf{P}_o^\kappa}{2} - \mathbf{P}_c^{\kappa^*} \quad (12)$$

where $\mathbf{P}_o^\kappa$ are the intersection of two adjacent lines selected from the $\mathrm{K}^3\{\kappa\}$, and $\mathbf{P}_a^\kappa$, $\mathbf{P}_b^\kappa$ are the end points of both wires,

respectively. Let us define $v_{\kappa,\kappa^*} = \mathbf{P}_c^{\kappa} - \mathbf{P}_c^{\kappa^*}$, we can obtain the constraints for the pitch and yaw

$$\begin{cases} \dfrac{v_a \cdot v_{\kappa,\kappa^*}}{||v_a||||v_{\kappa,\kappa^*}||} = \cos(\phi_{\kappa,\kappa^*}) \\ \dfrac{v_b \cdot v_{\kappa,\kappa^*}}{||v_b||||v_{\kappa,\kappa^*}||} = \cos(\varsigma_{\kappa,\kappa^*}) \end{cases} \text{for any visible } \mathrm{K}^3\{\kappa | \kappa \neq \kappa^*\}. \tag{13}$$

Fig. 3(b) shows an example of the constraints of $\mathrm{K}^3\{VI\}$ with $\kappa^* = IV$, with v_a, v_b, and $v_{VI,IV}$ being shown in the figure.

C. Similarity and Fitness Terms

Although the vehicles can be approximately tracked by the CMK tracking-based on the color information, the 3-D vehicle models cannot be properly fitted. To overcome this problem, we consider the fitness of the 3-D vehicle model while solving the optimization; in other words, the cost function in (4) becomes

$$J(\mathbf{x}) = \sum_{\kappa=1}^{N_k} w_\kappa \left(J_\kappa^s(\mathbf{x}) + J_\kappa^f(\mathbf{x}) \right) \tag{14}$$

where $J_\kappa^s(\mathbf{x})$ is the similarity term associated with the color similarity of the kernel, and $J_\kappa^f(\mathbf{x})$ is the fitness term associated with the fitness of the kernel, $\mathbf{x} \in \mathrm{R}^3$. Hence, the cost function in (5) can be rewritten as

$$\begin{aligned} \delta_{\mathbf{x}} &= \alpha\big(-\mathbf{I} + \mathbf{C}_{\mathbf{x}} \big(\mathbf{C}_{\mathbf{x}}^T \mathbf{C}_{\mathbf{x}}\big)^{-1} \mathbf{C}_{\mathbf{x}}^T \big) \mathbf{W} \big(\mathbf{J}_{\mathbf{x}}^s + \mathbf{J}_{\mathbf{x}}^f \big) \\ &\quad + \big(-\mathbf{C}_{\mathbf{x}} \big(\mathbf{C}_{\mathbf{x}}^T \mathbf{C}_{\mathbf{x}}\big)^{-1} \mathbf{C}(\mathbf{x}) \big) \\ &= \alpha\big(-\mathbf{I} + \mathbf{C}_{\mathbf{x}} \big(\mathbf{C}_{\mathbf{x}}^T \mathbf{C}_{\mathbf{x}}\big)^{-1} \mathbf{C}_{\mathbf{x}}^T \big) \mathbf{W} \mathbf{J}_{\mathbf{x}}^s \\ &\quad + \big(-\mathbf{C}_{\mathbf{x}} \big(\mathbf{C}_{\mathbf{x}}^T \mathbf{C}_{\mathbf{x}}\big)^{-1} \mathbf{C}(\mathbf{x}) \big) \\ &\quad + \alpha\big(-\mathbf{I} + \mathbf{C}_{\mathbf{x}} \big(\mathbf{C}_{\mathbf{x}}^T \mathbf{C}_{\mathbf{x}}\big)^{-1} \mathbf{C}_{\mathbf{x}}^T \big) \mathbf{W} \mathbf{J}_{\mathbf{x}}^f \\ &= \delta_{\mathbf{x}}^A + \delta_{\mathbf{x}}^B + \delta_{\mathbf{x}}^C \end{aligned} \tag{15}$$

where $\delta_{\mathbf{x}}^C$ is the movement vector associated with the fitness. Again it can be shown $\delta_{\mathbf{x}}^C$ is orthogonal to both $\delta_{\mathbf{x}}^A$ and $\delta_{\mathbf{x}}^B$ (see Appendix A), which implies moving along $\delta_{\mathbf{x}}^C$ is able to achieve better 3-D model fitting while maintaining the similarity and constraints of the kernels.

Given a 3-D kernel $\mathrm{K}^3\{\kappa\}$, the similarity term is determined by the similarity measurement, for instance, the distance between the target and the candidate kernels. As for the fitness term, the purpose is to further fit the 3-D vehicle model onto image. Since higher FES implies a better fitness of the model, the FESs of the kernels are considered as the measurement. Thus, we design a kernel density estimator by considering the FES. For a pixel $\mathbf{p}^\kappa$ belongs to $\mathrm{K}^2\{\kappa\}$, we calculate the FESs of the lines, associated with $\mathrm{K}^2\{\kappa\}$ centered at $\mathbf{p}^\kappa$ (annotated by $\mathrm{K}_{\mathbf{p}}^2\{\kappa\}$). In other words, the $\mathrm{K}_{\mathbf{p}}^2\{\kappa\}$ is projected from $\mathrm{K}^3\{\kappa\}$ which is shifted by $\overleftarrow{\mathbf{P}}^{\kappa} - \overleftarrow{\mathbf{P}}_c^{\kappa}$. Therefore, for each $\mathbf{p}^\kappa$, the FES $E_\kappa(\mathbf{p})$ is defined by

$$E_\kappa(\mathbf{p}) = -\sum_{l \in \mathrm{K}_{\mathbf{p}}^2\{\kappa\}} [\log(E_l)]. \tag{16}$$

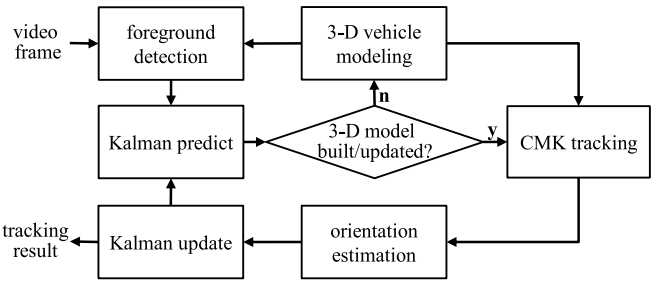


Fig. 4. Overall proposed system framework.

Note that $E_\kappa(\mathbf{p})$ is then applied to a kernel density estimator with kernel profile $k(\bullet)$. More specifically, $J_\kappa^f(\mathbf{x})$ is calculated as

$$J_\kappa^f(\mathbf{x}) = \frac{\sum_{i=1}^{n_\kappa} k\left(\left\| \frac{\mathbf{P}^\kappa - \overleftarrow{\mathbf{P}}_i^{\kappa}}{h'} \right\|^2 \right) E_\kappa(\mathbf{p}_i^\kappa)}{\sum_{i=1}^{n_\kappa} k\left(\left\| \frac{\mathbf{P}^\kappa - \overleftarrow{\mathbf{P}}_i^{\kappa}}{h'} \right\|^2 \right)}. \tag{17}$$

The purpose of the fitness term is to densely estimate the fitness of the kernel, such that the optimization tends to better fit the kernels. However, it is not necessary to evaluate all of $\{E_\kappa(\mathbf{p})\}$ for a kernel $\mathrm{K}^3\{\kappa\}$, since the effect of the fitness is dominated by the points around the center of $\mathrm{K}^3\{\kappa\}$. Thus, we uniformly sample the (dense) points $\{\mathbf{p}_i^\kappa\}_{i=1\ldots n_d}$, $1 \leq n_d \leq n_\kappa$, whose normalized distances between their back-projected points and the center of the kernel are smaller than a given threshold, that is, $|(\mathbf{P}^\kappa - \overleftarrow{\mathbf{P}}_i^{\kappa})/h'| < \rho, 0 < \rho \leq 1$.

IV. Complete Tracking System

We propose an automatic system which combines the Kalman-filtering framework with our 3-D vehicle model-based CMK tracking. Fig. 4 shows the schematics of the proposed vehicle tracking system. The first step is to segment the foreground objects by using background subtraction technique. Second, the Kalman prediction is applied to the segmented objects. Then, if the 3-D vehicle model is not yet built or needs to be updated, the predicted pose of the vehicle is used for building the 3-D vehicle model, which is then applied to the CMK tracking; otherwise, the CMK tracking is facilitated by the prebuilt 3-D vehicle model to track and locate the vehicles. Finally, the tracking results are used to update the states of the Kalman filter for further prediction.

After the CMK tracking, the kernels are local optimally shifted on the basis of the color similarity and shape fitness. However, when a vehicle turns left or right, the 3-D vehicle model cannot be fitted perfectly because of the orientation change. To improve the fitness of the 3-D vehicle model during the orientation change, we evaluate the FES of the 3-D vehicle models within a possible range of θ, and then select the 3-D vehicle model candidate with the highest FES. More specifically, for a tracked vehicle, we first predict the orientation (annotated by θ_{predict}) by Kalman-filtering. Then, the 3-D vehicle models with different θ are regarded as the candidates, where θ is set as $\theta_{\text{predict}} \pm \Delta\theta \cdot d$, $1 \leq d \leq 10$.

A typical value of $\Delta\theta$ is 0.5. Finally, the candidate with the highest FES is determined as the solution.

When occlusion occurs, the system requires different strategies to deal with partial and/or total occlusions. The challenge of partial occlusion is that the features of the tracked vehicle become unreliable in some partitions, resulting in mismatch of the tracked objects. In the CMK tracking, a less reliable partition (kernel) will be assigned by a smaller weight, thus the tracking tends to be more confident on a partition with larger weights [22]. Because of this adaptive weight assignment, the system has ability to overcome the partial occlusions during tracking. For the total occlusion, where the tracked object totally disappears, a possible way is to predict the traces of the vehicle and then resume the tracking when the vehicle appears again. Some approaches [5], [14] are proposed to deal with the total occlusion with reasonable performance. In our system, we believe that Kalman-filtering has ability to approximately predict the location of the totally occluded vehicles, because the vehicles usually go straight forward in a short time duration under the surveillance video. If the vehicle appears again after the total occlusion, Kalman-filtering is adopted to predict the location of the subsequent frame, and resume the tracking by the CMK tracking combined with the prebuilt 3-D vehicle model.

V. Experimental Results

Several experiments and corresponding discussions are provided in this section to demonstrate the performance of the proposed method. All the experiments are processed on a personal computer with a P4 2.67 GHz CPU and 2G DDR. The implementation is constructed by C/C++, and the experimental settings are described as follows. In the CMK tracking, K-L distance is used for all similarity measures, and the histogram of the object is constructed on the basis of the HSV color space with a roof kernel. In the 3-D vehicle modeling, the parameters used in $\text{EMNA}_{\text{global}}$ are $N = 2000$, $R = 100$, and the stopping threshold for the gradient magnitude is 2. For improving the robustness of the system, we rebuild the 3-D vehicle and update the kernels every five frames. The surveillance camera is supposed to be well calibrated, which implies intrinsic and extrinsic parameters are known. For the comparison, we manually labeled the ground truth of the 3-D vehicle models and measure the performance in terms of average errors (per frame), which is defined by

$$\text{err}_{\text{ave}}(t) = \frac{1}{N_t}\sum_{j=0}^{N_t}\sum_{i=0}^{15}\left\|x_i^j(t) - g_i^j(t)\right\| \tag{18}$$

where $x_i^j(t)$ is the 3-D location, $g_i^j(t)$ is the ground truth of the vertex i of the vehicle j, and N_t is the number of the tracked vehicles in the frame t. The testing datasets include PETS 2000 Dataset [40], portion of AVSS PV Easy [41], and our own recorded six datasets (Dataset 1–6). The content of the datasets are described in the left partition of the Table II. We use Dataset 4 for the discussion of the Sections V-A and V-B, and test all the videos under certain scenarios in the Section V-C. All the videos associated with the simulations reported in this paper can be viewed on our website.[1]

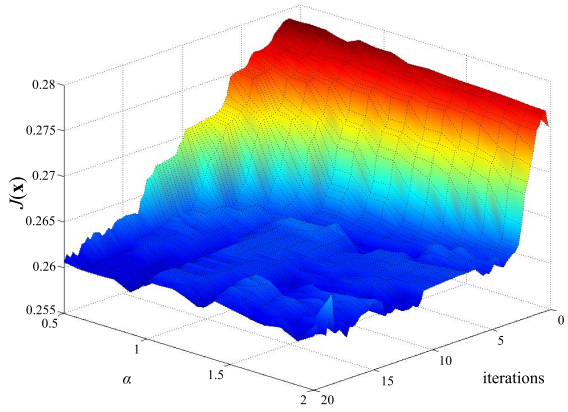


Fig. 5. Values of the $J(x)$ with different step size (a) and converging iterations, in case of 3-D kernels.

A. 2-D Kernel Versus 3-D Kernel

The 2-D kernels ($\text{K}^2\{\bullet\}$) is the projection of the 3-D kernels ($\text{K}^3\{\bullet\}$). Although they have different dimensions, the 3-D kernels still have some characteristics of the 2-D kernels.

1) The objects (patches in the 3-D model) are described by color histograms with a monotonically decreasing kernel profile $k(\bullet)$. The color information of $\text{K}^3\{\bullet\}$ is provided from the pixels within the corresponding $\text{K}^2\{\bullet\}$.
2) The mode of the kernel density in the local neighborhood can be found by applying the mean shift procedure [39]. Thus, we apply the mean shift vector with the opposite direction as our $\mathbf{J}_{\mathbf{x}}^s$ and $\mathbf{J}_{\mathbf{x}}^f$ in (15).
3) The $\text{K}^3\{\bullet\}$ are bound together under certain constraints, which are defined by the interrelationship of the kernels in 3-D space, so that these constraints can be consistent during the tracking.

Owing to these characteristics, the candidate kernels tend to be more similar to the target kernels, and the optimization in (2) can gradually converge. To speed up the convergence of the optimization in the 3-D CMK tracking, one possible way is to increase the step size α in (5). However, enlarging α will increase the probability of oscillation. Fig. 5 shows an example that how $J(\mathbf{x})$ changes with different α and different iterations. From the figure, when α is smaller, $J(\mathbf{x})$ decreases slower, taking more iterations to converge. On the contrary, when α is larger, $J(\mathbf{x})$ decreases more rapidly to relatively low-cost values. Nevertheless, $J(\mathbf{x})$ occurs with more oscillation before the convergence. For example, with $\alpha = 2$, $J(\mathbf{x})$ quickly decreases its value after four iterations, but still cannot converge at the 20th iteration. As for $1.0 < \alpha < 1.3$, the optimization converges in around 15 iterations, and there is no severe oscillation before the convergence. Thus, to achieve better performance, we empirically choose $\alpha = 1.2$ for the step size, and $\varepsilon_j = 0.26$, $T = 20$ for the stopping criterion. Besides, if the difference between the previous and the current

[1] website: http://allison.ee.washington.edu/kuanhuilee/3dvt

TABLE I
CONVERGENCE COMPARISON

Scheme	2DCMK	3DCMK−	3DCMK
Average iterations	9.561	11.744	15.915

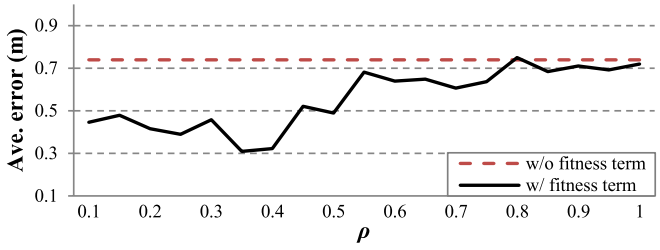


Fig. 6. Average errors of the 3-D CMK tracking with fitness term under different ρ.

ε_j is less than 0.001, the optimization is regarded as converged. Table I shows the average number of iterations required for the convergence in 2-D and 3-D CMK tracking, where 3-D CMK– and 3-D CMK denote the optimization without and with the fitness term, separately. As shown in the table, the optimization in 2-D CMK tracking takes less than ten iterations to converge, while the 3-D CMK tracking takes about 12–15 iterations. The reason why the 3-D CMK tracking requires more iterations to converge is that the movement vectors need to shift along three directions ($\delta_{\mathbf{x}}^A$, $\delta_{\mathbf{x}}^B$, and $\delta_{\mathbf{x}}^C$). The 3-D CMK tracking with fitness term ($J^f(\mathbf{x})$) needs more iterations than that without fitness term, which implies shifting along the $\delta_{\mathbf{x}}^C$ need more iterations to converge. Although including $J^f(\mathbf{x})$ takes more time, it is still acceptable and worthwhile with the improved tracking performance.

B. Fitness Term

In (14), the similarity term tends to retrieve a candidate kernel which has the highest similarity to the target kernel, however, only based on the color histogram. The fitness term plays an important role in achieving better fitness of the 3-D vehicle model on the image, because it tends to refine the location of the kernel for obtaining the best of the fitness level. As mentioned in Section III-C, a kernel density estimator is designed for fitting the vehicle 3-D model, and the dense points $\{\mathbf{p}_i^{\kappa}\}_{i\in 1,\ldots n_d}$ are uniformly sampled from the surrounding region $|(\mathbf{P}^{\kappa} - \overleftarrow{\mathbf{P}}_i^{\kappa})/h'| < \rho$. If ρ is too small, the moving range for refinement of the fitness will be limited; if ρ is too large, although the moving range becomes wide, the optimization may be stuck in a local minimum which is far away from the ground truth. Besides, sampling too many dense points for the pdf evaluation is not efficient, since many dense points do not have the color models highly similar to the target model. Fig. 6 shows the average errors of the 3-D CMK tracking with the fitness term in different ρ values. The solid line in Fig. 6 denotes the case of 3-D CMK with fitness term, and the dot line denotes the case of 3-D CMK without fitness term. In the simulations without the fitness term, we disregard the fitness term $\delta_{\mathbf{x}}^C$ in (15), with other elements being kept intact. When ρ becomes larger, the improvement of the fitness term becomes smaller, because many dense points far away from the ground truth dominate the movement to the other local minimums. On the contrary, the improvement also becomes smaller when ρ is smaller than 0.3. This is because there are too few reliable dense points to dominate the movement toward the better fitness. Owing to these observations, we choose $\rho = 0.4$ in the experiments to achieve better results.

C. Systematic Tracking Testing

In these experiments, we test our proposed approach under certain scenarios, including scale change, orientation change, and occlusion. The Dataset 1 is used for testing scale change, the Dataset 2 is for testing orientation change, the Dataset 3 is for testing occlusion, and the rest are for testing the situations of mixed scenarios. The influence of surrounding complexity is discussed in the Section V-C-4. The performance of the proposed 3-D CMK tracking approach (with and without fitness term) is also compared with other state-of-the-art intensity-based approaches, such as those of Zhang *et al.* [34] and Zheng and Peng [23]. All three approaches adopt the 3-D vehicle modeling in [34], but with different pose estimation schemes. Our implementations of their approaches do not include additional optimization, such as parallel computing mentioned in [34], to the 3-D vehicle modeling, since such optimization cannot change the accuracy of the pose estimation.

1) Scale Change: Scale change is one of the major issues during tracking. In [15], an adaptive scale change scheme for the kernel-based tracking is proposed. The basic idea is to adaptively change the bandwidth h of the kernel when the scale of the target varies, according to the scale invariance property of the similarity function of the kernel. The CMK tracking framework in [22] proposes a density estimator derived from [15], and performs well in human tracking. Because of the use of 3-D kernels, the movement vectors shifting in 3-D space are able to systematically adjust the scales of the kernels during the tracking. To demonstrate the efficiency of the 3-D CMK, we tested the vehicles in the Dataset 1, where the scale change happens to all vehicles (24 vehicles in total). Fig. 7 shows the visual results of the 3-D CMK in different frames. Table II shows the average errors of the tracking results with different approaches tested in Dataset 1. Fig. 10 shows the frame-by-frame average errors of a tracked vehicle with different approaches. The 3-D CMKs can achieve better results because the color information of the vehicle is effectively used.

To further investigate the performance between 2-D and 3-D CMK tracking, we compare them without updating the 3-D vehicle model. Fig. 11 shows the average similarity of the kernels of an example of the Dataset 1, in 2-D and 3-D CMK tracking. In Frame 56, the 3-D vehicle model is built, so that the average similarity is 1. In the 2-D CMK tracking, the average similarity continuously decreases owing to the scale change. However, in the 3-D CMK tracking, the average similarity decreases until Frame 67, and then maintains the values within a range. This is because the candidate kernels get modified when the scale is changing.

Fig. 7. Results of the 3-D CMK tracking in the Dataset 1, where the scale change happens to the vehicles.

TABLE II
AVERAGE ERROR (METER) EXPERIMENTS

Dataset	Situations			Edge Density	# vehicles in total	Approach				
	s	*o*	*c*			Zhang [34]	Zheng [23]	2DCMK	3DCMK–	3DCMK
Dataset 1	√			1.4137	24	0.7180	0.8058	0.6868	0.5824	**0.5668**
Dataset 2		√		0.9135	5	0.8386	0.8985	1.8892	0.7545	**0.7148**
Dataset 3			√	1.0022	2	0.5075	0.5838	0.4125	0.3996	**0.2209**
Dataset 4	√	√	√	1.3510	11	0.4136	0.4203	0.3553	0.3414	**0.2607**
Dataset 5	√	√		1.4604	12	0.6338	0.7109	0.5886	0.4963	**0.4575**
Dataset 6	√		√	5.6056	5	1.9771	2.3334	0.8174	0.7233	**0.6245**
PETS 2000	√	√		1.8940	3	0.8512	1.0732	0.7144	0.6976	**0.4616**
AVSS 2007	√	√		2.4921	7	0.6554	0.7277	0.7995	0.7786	**0.5043**

s: scale change, *o*: orientation change, *c*: occlusion; √ means the specific case occurs in the video.

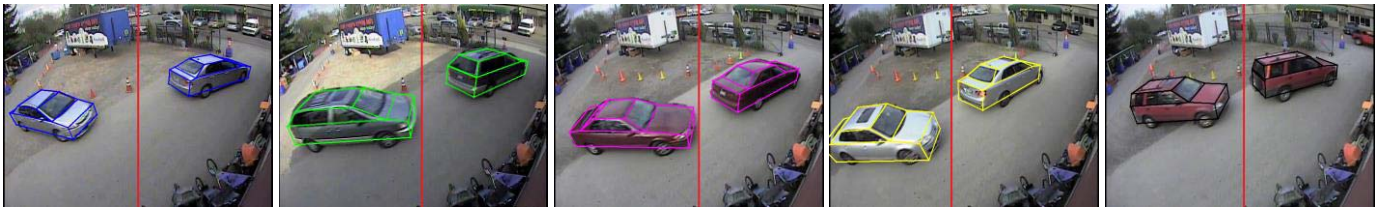


Fig. 8. Results of the 3-D CMK tracking in the Dataset 2, where the orientation change happens to the vehicles.

In our previous work [37], besides the 2-D CMK, we overcome the issue by updating the $K^2\{\bullet\}$ in a period of time to maintain the reliability of the $K^2\{\bullet\}$. Nevertheless, such scheme may not succeed in some cases, for example, when the vehicle is occluded with simultaneous scale change, or the 3-D vehicle model is not well fitted when the $K^2\{\bullet\}$ is rebuilt. These cases will cause inaccurate results, and the fitting errors further propagate during the tracking.

2) Orientation Change: Orientation change occurs when a vehicle turns left or right. To test the performance during the orientation change, we test the Dataset 2 that several sedans turn left in a corner. Fig. 8 shows the visual results of the 3-D CMK approach. Each figure is formed by two half frames: the left half is the first frame when a vehicle enters into the region of interest, and the other right half shows the tracked vehicle several frames after the first frame. The 3-D CMK performs well while the vehicles continuously change their orientations. Table II shows the average errors of the tracking results of different approaches, where the 3-D CMK approaches perform as well as the approaches of Zhang *et al.* [34] and Zheng and Peng [23]. This is because the performance of orientation change is dominated by the fitness evaluation. If the initial tracking contains adequate fitness in the iterative FES optimization, the CMK tracking cannot have obvious advantage. Fig. 12 shows the frame-by-frame average errors of a tracked vehicle with different approaches. The performance of the 3-D CMKs is as good as the comparing approaches. On the other hand, the results of 2-D CMK tracking are much worse than the other approaches, because it does not have ability to handle the orientation changes.

3) Occlusion: Occlusion is another major issue during tracking, because occlusion will break the matching of the object template previously described, especially in terms of 2-D geometry features. Owing to the adaptive weighting mechanism associated with the similarity, the 3-D CMK tracking is able to track the vehicles by facilitating with the more reliable kernels. In this section, we only consider the partial occlusion scenario. The total occlusion scenarios, where the whole vehicle is occluded such that it is impossible to accurately locate the vehicle, will require some additional dynamic state-space modeling (e.g., Kalman-filtering) to facilitate the tracking as discussed in Section IV. Fig. 9 shows the results of the proposed approach while one vehicle is partially occluded by another vehicle. Note that the vehicles can be well located and tracked without occlusion (see the leftmost, rightmost figures in Fig. 9) or with partial occlusion (see the middle three figures in Fig. 9). As the similarity of the occluded kernels becomes smaller, the smaller w_κ in (4) is assigned. Fig. 13 shows the frame-by-frame average errors of the tracking results with different approaches. From the figure, the 3-D CMK again performs better than the other approaches. While the occlusion

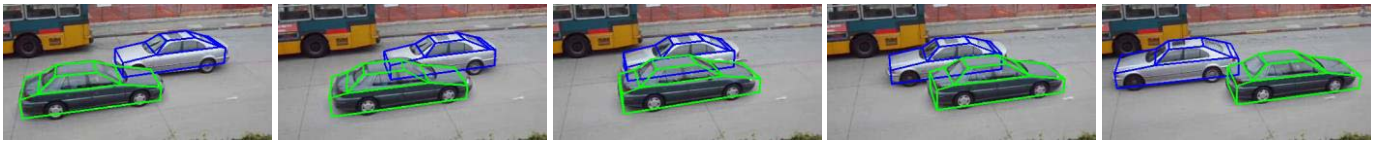


Fig. 9. Results of the 3-D CMK tracking in the Dataset 3, where a vehicle is occluded by another one.

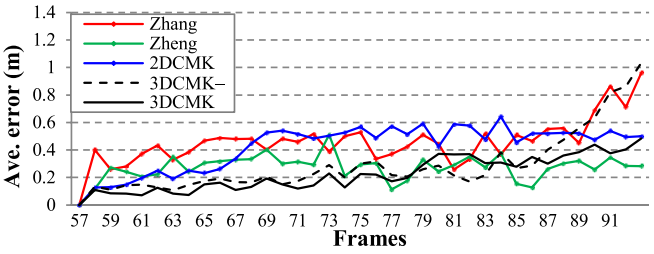


Fig. 10. Average errors of the tracking results with different approaches during scale change.

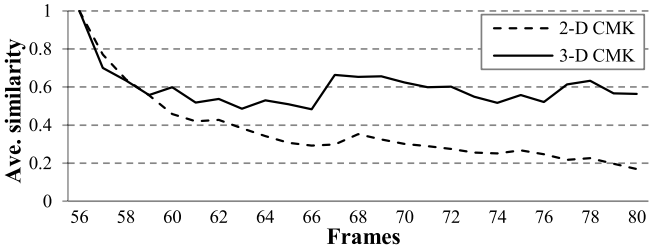


Fig. 11. Average similarity of the kernels of the example shown in Fig. 8.

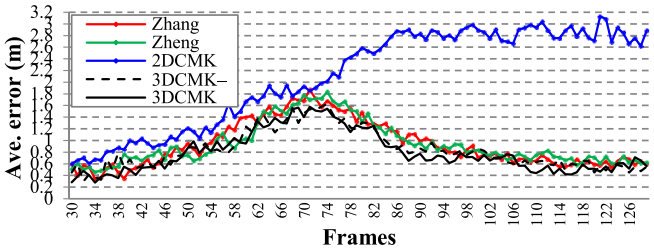


Fig. 12. Average errors of the tracking results with different approaches during orientation change.

occurs (from Frame 43 to Frame 60), the approaches of Zhang *et al.* [34] and Zheng and Peng [23] are not able to locate or track properly, because the gradients of the occluded vehicle are adversely affected by the occlusion. For the CMK-based tracking, they mainly rely on the nonoccluded partitions to maintain the proper tracking under the occlusion.

4) Automatic Tracking System: To show the performance of the proposed approach, we test the complete tracking system which combines the 3-D CMK tracking with the Kalman-filtering framework, as described in Section IV. Fig. 14(a)–(e) shows some of the results in the different datasets, respectively, showing the localization of different vehicles in different colors. We further compare the performance, in terms of the average errors, among several competing techniques, as shown in Table II. In the testing of PETS 2000, the proposed approach has obviously better

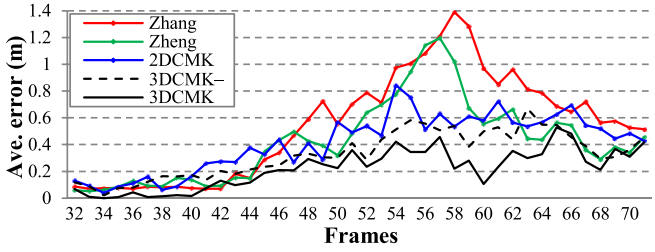


Fig. 13. Average errors of the tracking results with different approaches under (partial) occlusion.

performance because the approaches of Zhang *et al.* [34] and Zheng and Peng [23] evaluate the localization of the 3-D vehicles by only using the FES scheme, which is sensitive to the intensity changes of the surroundings. The 2-D CMK cannot achieve bigger improvement in presence of scale and orientation changes. In the AVSS 2007, most cases have scale changes and orientation changes, thus the 3-D CMK can achieve better performance than others. The results of the 2-D CMK and 3-D CMK– are worse than Zhang's approach, because both only rely on color information, resulting in the increased errors when the orientation changes occur. In the Dataset 1, Dataset 5, and Dataset 6, the scenarios are mostly scale changes, the CMK-based approaches perform slightly better than approaches of Zhang *et al.* [34] and Zheng and Peng [23]. The Dataset 2 is about the orientation change scenario, as shown in Fig. 8, and the Dataset 3 is about the occlusion scenario, as shown in Fig. 8, where the 3-D CMK performs better than other approaches. In the Dataset 4, the performance of the 3-D CMK is slightly improved because a few scale changes and occlusions occur. In the Dataset 6, total occlusion occurs when two vehicles intersect with each other [see the leftmost figure of Fig. 14(e)]. Since the background is complicated, the approaches of Zhang *et al.* [34] and Zheng and Peng [23] are adversely influenced by the initial misleading of the shape fitting.

From Table II, the 2-D CMK performs as along with the 3-D CMK–, which is mainly with regard to the color information. Therefore, the CMK-based approaches can achieve good results especially when there are image clutters caused by the surroundings; the improvement resulting from the fitness term is obvious, as it efficiently enhances the shape fitting of the 3-D vehicle models. Especially in the cases of the orientation changes, the improvement is more obvious than the other cases, since the projected 3-D vehicle models adapt more effectively.

5) Surrounding Complexity: Surrounding complexity has large impact on the fitness evaluation. Since higher complexity

Fig. 14. Vehicle in the PETS 2000 changes its orientation continuously. The results of the 3-D CMK tracking in (a) PETS 2000, (b) AVSS 2007, (c) Dataset 4, (d) Dataset 5, and (e) Dataset 6.

may produce higher FES, there is higher possibility to lead the FES evaluation to the unexpected local optima. To measure the surrounding complexity, we use edge density in the region of interest (ROI) as the metric

$$d_{\text{edge}} = \frac{1}{A_{\text{roi}}} \sum_{\mathbf{p} \in \text{ROI}} e(\mathbf{p}) \tag{19}$$

where $e(\mathbf{p})$ is the magnitude of the Sobel 3×3 operation at $\mathbf{p}$, and A_{roi} is the area of the ROI.

Higher d_{edge} normally indicates higher scene variability, which can produce possible local maxima of the FES, even not the vehicle. If the FES evaluation traps in an unexpected local optima, the error of the shape fitting gradually increases, and then the predictions of the tracking is further degraded. On the other hand, the 3-D CMKs track the object by first considering color information, thus provides better initialization for the FES optimization. As shown in the Table II, the 3-D CMK tends to have more obvious improvement than other approaches when testing the Datasets with higher d_{edge}, such as Datasets 5, 6, and PETS2000. For AVSS2007, the performance is only slightly improved. This is because the vehicles in that video are more discriminative when compared with the surroundings. In the occlusion case, such as Dataset 2, the fitting confusion is caused by the occlusion. If the edge density of the occlusion is high, the proposed approach can achieve better performance; otherwise, if the edge density of the occlusion is low, such as the traffic lights in the Dataset 4, the improvement is not as impressive.

D. Computation

Because of the high efficiency of the kernel-based tracking, the proposed approach is quite efficient in computations. To demonstrate the efficiency, we compare the CPU time consumption of the pose estimation, that is, locating the pose

TABLE III
COMPUTATIONAL CONSUMPTION

Ave. time (second)	Approach				
	Zhang [34]	Zheng [23]	2DCMK	3DCMK–	3DCMK
per vehicle	47.268	5.277	1.197	1.283	1.425

parameters (X', Y', θ), in different approaches. Table III shows the results of the computational time in different approaches, in terms of seconds per vehicle. The approach of Zhang *et al.* [34] has the highest computation owing to large amount of generations of the individuals in the $\text{EMNA}_{\text{global}}$ framework. Instead of generating the individuals randomly, the approach of Zheng and Peng [23] efficiently narrows the searching range, so that the computation is greatly reduced. Within the CMK-based approaches, the 2-D CMK has the best results because the optimization converges along the 2-D mean-shift vector. For the 3-D CMK– and 3-D CMK, the optimization converges along the 3-D mean-shift vector, which implies that more iterations are needed. As mentioned in Section V-A, when compared with 2-D CMK, 3-D CMK– needs about two more iterations on average, and 3-D CMK needs about six more iterations on average. From the Table III, the 3-D CMK– spends one more second on average, and the 3-D CMK is about two more seconds on average; which is still relatively more efficient than the approaches of Zhang *et al.* [34] and Zheng and Peng [23].

E. Limitations

Although performing well in many aforementioned scenarios, the proposed approach has some limitations. First, the 3-D vehicle modeling plays an important role in initially describing the kernels, as well as the target kernels. The tracking procedure is to find the candidate kernels which have the highest similarity to the corresponding target kernel. If the 3-D vehicle modeling cannot provide the reliable target kernels, the tracking will be based on the incorrect kernels, and cause the error propagation. Second, the color of the vehicle also impacts the tracking results, because 3-D CMK tracking is highly dependent on the color histogram. For example, if a vehicle is occluded by an obstacle which has very similar color to the occluded vehicle, the CMK tracking will be confused and create the wrong tracking results.

F. Discussion

In the approach of Zhang *et al.* [34], the 3-D vehicle models are located by iteratively optimizing its parameters, according to the measurement of the fitness in terms of intensity and gradients. The approach of Zheng and Peng [23], also based on intensity, efficiently measures the fitness of the 3-D vehicle models in certain ranges, so as to reduce the searching computation. The difference between these intensity-based approaches and our proposed approach is that we further consider the color information. In our proposed framework, the optimization of the cost function includes not only the color histogram but the 3-D shape fitness score. Therefore, the vehicles are tracked based on the color, which is a distinctive characteristic of the vehicle, and their shapes are fitted with 3-D models at the same time. Another difference between these competing approaches and our proposed approach is the optimization of the cost function. Their approaches try to search the location of the vehicles with a relatively better fitness level, which implies that high computation is necessary to achieve good performance. However, the proposed approach optimizes the cost function along the mean-shift vectors, which is able to quickly converge.

Most importantly, we propose a 3-D model-based tracking framework for vehicle tracking. Theoretically, the framework can also be applied to a highly complicated 3-D vehicle model [35], [36], or any object with 3-D model, to efficiently track and locate the objects.

VI. CONCLUSION

A novel model-based vehicle localization approach under a surveillance camera is proposed in this paper. Our contribution is to propose a vehicle tracking framework, which efficiently locates and tracks the vehicles by combining the CMK tracking with a 3-D deformable vehicle model. By applying 3-D movement of the kernels, the CMK tracking can achieve better performance. Moreover, a kernel density estimator designed for refining the fitness of the 3-D vehicle model is considered in the cost function, so that the CMK tracking tracks the vehicles according to not only the similarity of the color information but also the fitness of the kernels. Experimental results demonstrate the robustness, effectiveness, and reliability of the proposed approach. Although the proposed approach is applied to the vehicle tracking, the concept of the framework can be further applied to the various kinds of the object tracking.

APPENDIX A

In [22], the fact that the $\delta_{\mathbf{x}}^A$ and the $\delta_{\mathbf{x}}^B$ are orthogonal to each other is proved. The following proves the $\delta_{\mathbf{x}}^C$ is orthogonal to both the $\delta_{\mathbf{x}}^A$ and the $\delta_{\mathbf{x}}^B$:

$$
\begin{aligned}
&\left(\delta_{\mathbf{x}}^B\right)^T\left(\delta_{\mathbf{x}}^C\right)\\
&= \left(-\mathbf{C}_{\mathbf{x}}\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\mathbf{C}(\mathbf{x})\right)^T\left(\alpha\left(-\mathbf{I}+\mathbf{C}_{\mathbf{x}}\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\mathbf{C}_{\mathbf{x}}^T\right)\mathbf{J}_{\mathbf{x}}^s\right)\\
&= \alpha\left(-\mathbf{C}(\mathbf{x})^T\left(\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\right)^T\mathbf{C}_{\mathbf{x}}^T\right)\left(-\mathbf{J}_{\mathbf{x}}^s+\mathbf{C}_{\mathbf{x}}\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^s\right)\\
&= \alpha\left(\mathbf{C}(\mathbf{x})^T\mathbf{C}_{\mathbf{x}}^{-1}\mathbf{J}_{\mathbf{x}}^s-\mathbf{C}(\mathbf{x})^T\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^s\right)\\
&= \alpha\mathbf{C}(\mathbf{x})^T\left(\left(\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\right)^T-\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\right)\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^s\\
&= 0.
\end{aligned}
$$

Thus, the $\delta_{\mathbf{x}}^A$ and the $\delta_{\mathbf{x}}^B$ are orthogonal to each other

$$
\begin{aligned}
&(\delta_{\mathbf{x}}^A)^T(\delta_{\mathbf{x}}^C)\\
&\quad= \left(\alpha\left(-\mathbf{I}+\mathbf{C}_{\mathbf{x}}\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\mathbf{C}_{\mathbf{x}}^T\right)\mathbf{J}_{\mathbf{x}}^s\right)^T\\
&\qquad\cdot\left(\alpha\left(-\mathbf{I}+\mathbf{C}_{\mathbf{x}}\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\mathbf{C}_{\mathbf{x}}^T\right)\mathbf{J}_{\mathbf{x}}^f\right)\\
&\quad= \alpha^2\left(-\left(\mathbf{J}_{\mathbf{x}}^s\right)^T+\left(\mathbf{J}_{\mathbf{x}}^s\right)^T\mathbf{C}_{\mathbf{x}}\left(\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\right)^T\mathbf{C}_{\mathbf{x}}^T\right)\\
&\qquad\cdot\left(-\mathbf{J}_{\mathbf{x}}^f+\mathbf{C}_{\mathbf{x}}\left(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\right)^{-1}\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^f\right)
\end{aligned}
$$

$$
\begin{aligned}
&= \alpha^2\big(\big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\big(\mathbf{J}_{\mathbf{x}}^{f}\big) - \big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\mathbf{C}_{\mathbf{x}}\big(\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\big)^T\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^{f} \\
&\quad -\big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\mathbf{C}_{\mathbf{x}}\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^{f} \\
&\quad +\big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\mathbf{C}_{\mathbf{x}}\big(\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\big)^T\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^{f}\big) \\
&= \alpha^2\big(\big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\big(\mathbf{J}_{\mathbf{x}}^{f}\big) - \big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\mathbf{C}_{\mathbf{x}}\big(\big(\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\big)^T + \big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1} \\
&\quad -\big(\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\big)^T\big)\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^{f}\big) \\
&= \alpha^2\big(\big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\big(\mathbf{J}_{\mathbf{x}}^{f}\big) - \big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\mathbf{C}_{\mathbf{x}}\big(\mathbf{C}_{\mathbf{x}}^T\mathbf{C}_{\mathbf{x}}\big)^{-1}\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^{f}\big) \\
&= \alpha^2\big(\big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\big(\mathbf{J}_{\mathbf{x}}^{f}\big) - \big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\mathbf{C}_{\mathbf{x}}\big(\mathbf{C}_{\mathbf{x}}\big)^{-1}\big(\mathbf{C}_{\mathbf{x}}^T\big)^{-1}\mathbf{C}_{\mathbf{x}}^T\mathbf{J}_{\mathbf{x}}^{f}\big) \\
&= \alpha^2\big(\big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\big(\mathbf{J}_{\mathbf{x}}^{f}\big) - \big(\mathbf{J}_{\mathbf{x}}^{s}\big)^T\big(\mathbf{J}_{\mathbf{x}}^{f}\big)\big) \\
&= 0.
\end{aligned}
$$

Therefore, the $\delta_{\mathbf{x}}^{A}$ and the $\delta_{\mathbf{x}}^{C}$ are orthogonal to each other.

REFERENCES

[1] 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.

[2] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," in *Proc. Eur. Conf. Comput. Vis.*, May 1994, pp. 189–196.

[3] D. M. Ha, J. M. Lee, and Y. D. Kim, "Neural-edge-based vehicle detection and traffic parameter extraction," *J. Image Vis. Comput.*, vol. 22, no. 11, pp. 899–907, 2004.

[4] P. L. M. Bouttefroy, A. Bouzerdoum, S.-L. Phung, and A. Beghdadi, "Vehicle tracking by non-drifting mean-shift using projective Kalman filter," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 61–66.

[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] Y. Wang, "Real-time moving vehicle detection with cast shadow removal in video based on conditional random field," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 437–441, Mar. 2009.

[7] J. W. Hsieh, S. H. Yu, Y. S. Chen, and W. F. Hu, "Automatic traffic surveillance system for vehicle tracking and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 175–187, Jun. 2006.

[8] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 259–289, May 2008.

[9] J.-Y. Choi, K.-S. Sung, and Y.-K. Yang, "Multiple vehicles detection and tracking based on scale-invariant feature transform," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Sep. 2007, pp. 528–533.

[10] N. Buch, J. Orwell, and S. A. Velastin, "Three-dimensional extended histograms of oriented gradients (3-D HOG) for classification of road users in urban scenes," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2009.

[11] X. Cao, C. Wu, J. Lan, and P. Yan, "Vehicle detection and motion analysis in low-altitude airborne video under urban environment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1522–1533, Oct. 2011.

[12] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. Eur. Conf. Comput. Vis.*, May 2002, pp. 661–675.

[13] T. Xiong and C. Debrunner, "Stochastic car tracking with line- and color-based features," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 324–328, Dec. 2004.

[14] J. Scharcanski, A. B. Oliveira, 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.

[15] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[16] V. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 176–183.

[17] A. Yilmaz, "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–6.

[18] A. Tyagi, M. Keck, J. W. Davis, and G. Potamianos, "Kernel-based 3D tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[19] M. Kushwaha and X. Koutsoukos, "3D target tracking in distributed smart camera networks with in-network aggregation," in *Proc. ACM/IEEE Int. Conf. Distrib. Smart Cameras*, Sep. 2010, pp. 25–32.

[20] Z. Fan, Y. Wu, and M. Yang, "Multiple collaborative kernel tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 502–509.

[21] J. Fang, J. Yang, and H. Liu, "Efficient and robust fragments-based multiple kernels tracking," *Int. J. Electron. Commun.*, vol. 65, no. 11, pp. 915–923, Nov. 2011.

[22] C.-T. Chu, J.-N. Hwang, H.-I. Pai, and K.-M. Lan, "Tracking human under occlusion based on adaptive multiple kernels with projected gradients," *IEEE Trans. Multimedia*, vol. 5, no. 7, pp. 1602–1615, Nov. 2013.

[23] Y. Zheng and S. Peng, "Model based vehicle localization for urban traffic surveillance using image gradient based matching," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 945–950.

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

[25] M. Haag and H.-H. Nagel, "Combination of edge element and optical flow estimates for 3D-model-based vehicle tracking in traffic image sequences," *Int. J. Comput. Vis.*, vol. 35, no. 3, pp. 295–319, Dec. 1999.

[26] J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank, "3-D model-based vehicle tracking," *IEEE Trans. Image Process.*, vol. 4, no. 10, pp. 1561–1569, Oct. 2005.

[27] J. Liebelt, C. Schmid, and K. Schertler, "Viewpoint-independent object class detection using 3D feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[28] C. Reinbacher, M. Ruther, and H. Bischof, "Pose estimation of known objects by efficient silhouette matching," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1080–1083.

[29] N. Buch, J. Orwell, and S. A. Velastin, "Urban road user detection and classification using 3D wire frame models," *IET Comput. Vis.*, vol. 4, no. 2 pp. 105–116, Jun. 2010.

[30] K. S. Brisdon, "Hypothesis verification using ICONIC matching," Ph.D. dissertation, Dept. Elect. Eng., Univ. Reading, Berkshire, U.K., 1990.

[31] J. Ferryman, A. Worrall, G. Sullivan, and K. Baker, "A generic deformable model for vehicle recognition," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 1995, pp. 127–136.

[32] T. N. Tan, G. D. Sullivan, and K. D. Baker, "Model-based localisation and recognition of road vehicles," *Int. J. Comput. Vis.*, vol. 27, no. 1, pp. 5–25, Mar. 1998.

[33] T. N. Tan and K. D. Baker, "Efficient image gradient based vehicle localization," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1343–1356, Aug. 2000.

[34] Z. Zhang, T. Tan, K. Huang, and Y. Wang, "Three-dimensional deformable-model-based localization and recognition of road vehicles," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 1–13, Jan. 2012.

[35] M. J. Leotta and J. L. Mundy, "Vehicle surveillance with a generic, adaptive, 3D vehicle model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1457–1469, Jul. 2011.

[36] Y.-L. Lin, M.-K. Tsai, W. H. Hsu, and C.-W. Chen, "Investigating 3-D model and part information for improving content-based vehicle retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 3, pp. 401–412, Mar. 2013.

[37] K.-H. Lee, J.-N. Hwang, J. Yu, and K.-Z. Lee, "Vehicle tracking iterative by Kalman-based constrained multiple-kernel and 3-D model-based localization," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2013, pp. 2396–2399.

[38] P. Larrauaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Norwell, MA, USA: Kluwer, 2001.

[39] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[40] *PETS Data Base* [Online]. Available FTP: ftp://ftp.pets.rdg.ac.uk/pub/PETS2000/, accessed 2000.

[41] *Advanced Video and Signal based Surveillance (AVSS)* [Online]. Available: http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html, accessed 2007.

Kuan-Hui Lee received the B.S. degree from the Department of Electrical Engineering, National Taiwan Ocean University, Keelung, Taiwan, in 2003 and the M.S. degree from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 2005. He is currently working toward the Ph.D. degree with the Information Processing Laboratory, Department of Electrical Engineering, University of Washington, Seattle, WA, USA.

He was with HTC Corporation, Taoyuan, Taiwan, where he developed multimedia applications for smart phones from 2007 to 2009. His research interests include computer vision, machine learning, and video/image processing.

Jenq-Neng Hwang (F'01) received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree from University of Southern California, Los Angeles, CA, USA.

He joined the Department of Electrical Engineering, University of Washington, Seattle, WA, USA, in 1989, where he was promoted to Full Professor in 1999. He is currently the Associate Chair of Research with the Department of Electrical Engineering. He has written more than 300 journals, conference papers, and book chapters in the areas of multimedia signal processing, and multimedia system integration and networking, including an authored textbook entitled *Multimedia Networking: From Theory to Practice* (Cambridge University Press). He has a close working relationship with the industry on multimedia signal processing and multimedia networking.

Dr. Hwang is a Founding Member of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society, and was the Society's Representative to the IEEE Neural Network Council from 1996 to 2000. He is a member of the Multimedia Technical Committee of the IEEE Communication Society and the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society. He served as an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON IMAGE PROCESSING, and *IEEE Signal Processing Magazine*. He is on the editorial board of *ETRI Journal*, *International Journal of Digital Multimedia Broadcasting*, and *Journal of Signal Processing Systems*. He was the Program Co-Chair of the 1998 International Conference on Acoustics, Speech and Signal Processing and the 2009 International Symposium on Circuits and Systems. He received the 1995 IEEE Signal Processing Society's Best Journal Paper Award.

Shih-I Chen received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1986 and the Ph.D. degree in electrical engineering from University of Maryland, College Park, MD, USA, in 1996.

His research interests include wireless communications and multimedia. He joined Motorola, Chicago, IL, USA, in 1997, where he was responsible for the software simulation of mobile communications system for performance validation and optimization. He is currently with Institute for Information System, Taipei, where he leads the development work of broadband wireless systems such as WiMAX, LTE, and beyond-4G mobile communications.

Dr. Chen has been the primary investigator of several government-funded research projects in 3G and 4G mobile communications software and systems, and received multiple distinguished project awards.