A Novel On-Road Vehicle Detection Method Using πHOG

Jisu Kim, Jeonghyun Baek, and Euntai Kim

***Abstract*—In this paper, a new on-road vehicle detection method is presented. First, a new feature named the Position and Intensity-included Histogram of Oriented Gradients (PIHOG or πHOG) is proposed. Unlike the conventional HOG, πHOG compensates the information loss involved in the construction of a histogram with position information, and it improves the discriminative power using intensity information. Second, a new search space reduction (SSR) method is proposed to speed up the detection and reduce the computational load. The SSR additionally decreases the false positive rate. A variety of classifiers, including support vector machine, extreme learning machine, and k-nearest neighbor, are used to train and classify vehicles using πHOG. The validity of the proposed method is demonstrated by its application to Caltech, IR, Pittsburgh, and Kitti datasets. The experimental results demonstrate that the proposed vehicle detection method not only improves detection performance but also reduces computation time.**

***Index Terms*—Vehicle detection, feature, HOG, search space reduction, Bayesian approach, sliding-window approach, SVM.**

I. Introduction

OVER the past decade, on-road vehicle detection systems have received considerable attention and have emerged as a key issue in intelligent vehicle (IV) realms. In particular, vehicle detection using only a monocular camera is an attractive option; however, it is considered a challenging task because of the various types of vehicles and complicated backgrounds encountered while driving. Many previous works have addressed vehicle detection using a monocular camera [1]–[5]. Most of these methods consist of two steps: hypothesis generation (HG) and hypothesis verification (HV) [1]. In HG, possible vehicle candidates are selected from an image using (usually) low-computation methods. In HV, the candidates selected in HG are tested using a strong classifier to verify the presence of the vehicles.

To date, various HG approaches have been reported and can be divided into the following three categories [1]: 1) knowledge-based, 2) stereo-based, and 3) motion-based. In knowledge-based approaches, a priori knowledge about the vehicles is exploited to select the vehicle candidates in an image. The examples of the knowledge used here are symmetry [6], [7], color [3], [8], shadow [9], [10], geometrical features such as corners or horizontal/vertical edges [11], and vehicle lights [12]. In stereo-based approaches, the depth or disparity difference between the foreground and background is used to predict vehicle presence. There are two types of stereo-based methods: one uses the disparity map [13] and the other uses inverse perspective mapping (IPM) [14]. In motion-based approaches, optical flows are used to localize the candidate vehicles [15], [16]. The key difficulty arising in these motion-based approaches is to distinguish the flow caused by the preceding vehicles from the flow caused by ego motion.

Some research works without HG have been reported in vehicle or pedestrian detection [17], [18]. These studies implemented exhaustive search (ES) using a sliding window method and omit the HG. The motivations are clear and can be summarized as

1) If HG misses the vehicles, the strong classifier in HV cannot be used. This is a serious problem;
2) HG can be time-consuming and might not be as fast as expected;
3) Some interesting techniques have been reported for the computation reduction of a sliding window method; for example, integral histogram [19], fast feature pyramids [20] and search space reduction (SSR) [10], [21]–[27];
4) With the development of computer hardware such as GPU, the sliding window method is considered to be more reliable than HG.

The sliding window approach is widely used in ES and scans the whole image in an exhaustive manner while changing the position and size (scale) of the window. The sliding-window approach, however, is somewhat inefficient in that the whole image is scanned in the vehicle search, including the sky or building regions, which slows detection. To overcome this problem and speed up detection, several methods have been reported. These methods can be classified into two groups: 1) knowledge-based sliding-window approaches [21]–[24] and 2) coarse-to-fine approaches [10], [25]–[27]. In [21], an adaptive sliding-window method based on prior knowledge, such as camera pose, distance, and object size, is used. It is assumed that the camera position and pitch angle are already known. The vehicle width is also assumed to range from 1.5 to 2.0 m. Using this information, image patch sizes are defined according to distance. This improves operation time by removing unlikely image parts. In [22], the concept of the adaptive sliding window is used in a way similar to [21]. However, [22] uses geometric

Manuscript received June 24, 2014; revised December 11, 2014 and June 8, 2015; accepted July 1, 2015. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology under NRF-2013R1A2A2A01015624. The Associate Editor for this paper was P. Grisleri.

The authors are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Korea (e-mail: jisukim2000@yonsei.ac.kr; jhyun25@yonsei.ac.kr; etkim@yonsei.ac.kr).

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

Digital Object Identifier 10.1109/TITS.2015.2465296

constraints for the adaptive sliding-window approach and so is more robust than the method in [21]. Geronimo *et al.* present a method that defines image patches by fitting a road surface and estimating the camera pose [23]. Therefore, the candidate is generated by the corresponding depth. Moreover, the RANSAC algorithm is used to reduce 3D outliers and establish a linear fit for a road surface. In [24], the linear model between vehicle size and vehicle position is estimated using the recursive least square method. In addition, the linear model is updated according to the detected result. None of the above algorithms learns the relation between car size and car position or the associated online uncertainty using the learning theory.

The coarse-to-fine approach has also been widely used. In [25], the authors use a two-stage approach in video sequences. They consider the advantages of both the Haar-like wavelet and HOG features. The Haar-like feature requires low computation time; however, it leads to many false positives. Therefore, this feature is suitable for generating hypotheses in the first stage. On the other hand, the HOG feature involves high computation time and a high detection rate. For this reason, it is suitable for verifying hypotheses in the second stage. In [26], the coarse-to-fine approach is also used for discriminative multi-resolution with a multi-stage classifier according to resolution. Using low-resolution features enables detection of coarse regions of the object. Within those regions, high-resolution features enable detection of fine regions of the object. In [27], the statistical search is implemented to estimate the likelihood density of the candidate using Monte Carlo sampling. To detect the object, a multi-stage particle is used. This method has high detection performance; however, it incurs high computation time. In [10], vehicle and road regions are generated using lane detection. In these regions, effective candidates are generated using vehicle shadow and tire features. In general, the coarse-to-fine approach takes longer than the knowledge-based sliding window approach.

After HG/ES/ES with SSR, HV should be conducted. In HV, two points should be considered: 1) which features will be extracted from a hypothesized window and 2) which classifier is used to test the hypothesis. Concerning extraction and evaluation of a feature from a window, several features have been reported. The Haar-like wavelet [17], histogram of oriented gradient (HOG) [28], and local binary pattern (LBP) [29] are the most popular such features. The Haar-like wavelet is simpler than the HOG feature and requires less evaluation time. The HOG feature requires a longer time but is more robust in vehicle detection than is the Haar-like feature. These features were combined to improve detection performance in [30]. The LBP is also commonly used for vehicle detection [29]. However, the LBP is so sensitive to local intensity variations that different patterns can be generated from identical vehicles. To overcome this problem, the Local Gradient Pattern (LGP) method is proposed in [31]. The LGP generates constant patterns irrespective of local intensity variations. Further, many variants of HOG have been proposed to improve the discriminant power; for example, Local Structured HOG (LSHOG) [32]. The original HOG has difficulty effectively describing the local structure; LSHOG solves this problem by adopting the local gradient energy and capturing local structure [32].

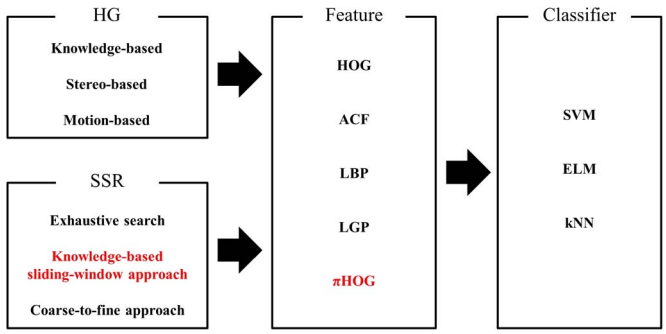


Fig. 1. Taxonomy graph of the vehicle detection methods.

Unfortunately, however, the above features [28]–[32] have two problems: 1) they use blocks or cells to construct histograms, during which valuable information is lost. A detailed example is given in Section III. 2) Although intensity is also a powerful clue for vehicles, most of the above features use only the gradient of the window, except ACF and ICF. To solve these two problems, a new feature called the Position and Intensity-included Histogram of Oriented Gradients (PIHOG or πHOG) is proposed in this paper. The position information about the region in which a histogram occurs is used to compensate for the information loss experienced during histogram construction. Further, the regions in which the intensity of vehicle images does not change much are identified and used as features in πHOG. The intensity information results in improved discriminative power in πHOG. The way in which the intensity is used in πHOG is completely different from the ways in which it is used in ICF and ACF. In ACF and ICF, features are sums of pixels in each block of the intensity image; thus, all the pixels in the entire intensity image are used. In πHOG, however, the intensity values of the pixels only in a certain region (which will be explained later) are formatted as standard normal deviates and used.

With regard to classifiers that can be applied to vehicle detection, the most popular candidates are AdaBoost [33], support vector machines (SVMs) [34], extreme learning machine (ELM) [35], k-nearest neighbor (kNN) [36], and neural networks (NNs) [37]. In particular, the Haar-like feature is often used with AdaBoost [21], while HOG is combined with SVM [38]. Recently, Soft-Cascade [39], Deformable Part Model (DPM) [18], and Deep Learning [40] have received attention within the IV society.

In this paper, we focus on new features for vehicle detection and subsequent search space reduction (SSR). The proposed method can be positioned in a taxonomy graph, as shown in Fig. 1, where the proposed method is marked in red. A new feature, πHOG, is proposed to improve vehicle detection accuracy, and efficient SSR is produced by exploiting the relationship between the position and size of the vehicle. The classifier is not addressed in this paper.

The remainder of this paper is organized as follows. In Section II, the proposed vehicle detection method is briefly introduced. In Section III, the proposed πHOG feature is explained, while the proposed SSR method is outlined in Section IV. Experimental results comparing the proposed

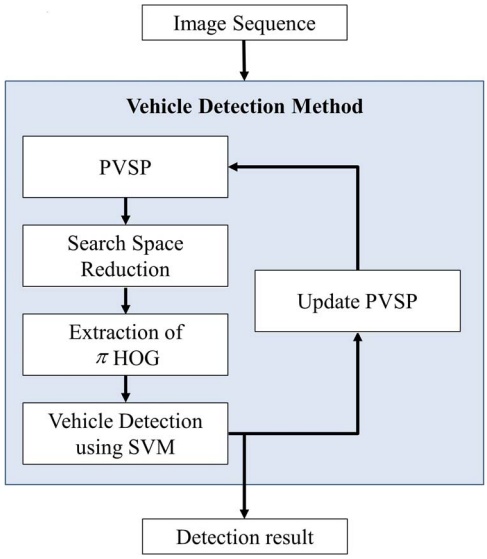


Fig. 2. Overview of the proposed vehicle detection method.

method with previous methods are described in Section V. The conclusions of this paper are given in Section VI.

II. System Overview

The objective of the proposed method is to detect vehicles in urban road environments. The common urban environment includes a variety of buildings, traffic signs, or pedestrians as background. The proposed vehicle detection method consists of four steps: 1) formation of a Position-wise Vehicle Size Predictor (PVSP) and its update, 2) SSR using the PVSP, 3) extraction of πHOG while sliding a window in the PVSP, and 4) vehicle detection using SVM, then returning to the first step. Fig. 2 presents an outline of the proposed vehicle detection method.

In Step 1, the region in which vehicles are likely to be detected is modeled as the PVSP, which represents the correlation between vehicle position and size. If it is not the first iteration, the process is updated using the detections of the previous iteration. In Step 2, the search space is refined and narrowed using the PVSP. The SSR leads to both an increase in detection speed and a decrease in false detection. In Step 3, a window slides within the reduced search space, and a new feature πHOG is computed for each window, as shown in Fig. 3. πHOG consists of three parts: position, intensity, and conventional HOG. The position part of πHOG specifies the orientation of the histogram and compensates for the limitations of the simple histogram. The intensity part of πHOG utilizes the intensity information of the window and extracts the region unique to the vehicles. These two parts improve the discriminative power of the proposed πHOG. In Step 4, each window is tested using πHOG and the SVM. Instead of a kernel SVM, a linear SVM is employed for real-time implementation. When a vehicle is detected, the PVSP is updated using a Bayesian approach.

III. πHOG: A New Feature

The HOG feature is very popular in the computer vision realm and is widely used in a variety of applications, including

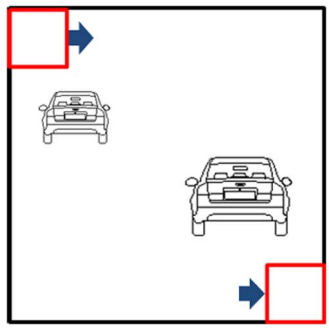


Fig. 3. Example of the sliding window.

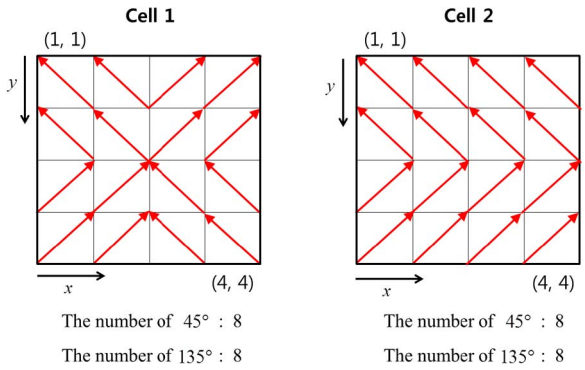


Fig. 4. Two different cells with the same HOG.

vehicle and pedestrian detection. The HOG feature, however, has some drawbacks.

1) HOG is actually a histogram, in which the positions of the gradients are lost. Thus, two completely different cells can have the same histogram. This idea is explained in Fig. 4. As shown in the figure, a cell consists of 16 pixels, and each pixel has a gradient vector denoted by a red arrow. The positions of the upper left and lower right pixels are (1, 1) and (4, 4), respectively. For the sake of simplicity, it is assumed that all gradients have the same intensities. Obviously, the two cells have different gradients; however, they have the same histogram of oriented gradients because both cells have the same number of 45° and 135° components. Shown in Fig. 5 is the actual case in which HOG does not provide discriminative information about the vehicle and background. As shown in the figure, four cells are used to build a HOG feature. It is evident that the vehicle window in the first row and the background window in the second row are quite different from each other; nevertheless, they have very similar HOG features.
2) Only edge information of the image is used; intensity information is not at all utilized in HOG. Thus, it is apparent that edge-based HOG might not be sufficient for vehicle detection. Thus, there remains further opportunity for improved vehicle detection performance using image intensity.

By combining the above two ideas, the new πHOG feature is realized. The πHOG position and intensity parts are explained in the following subsections.

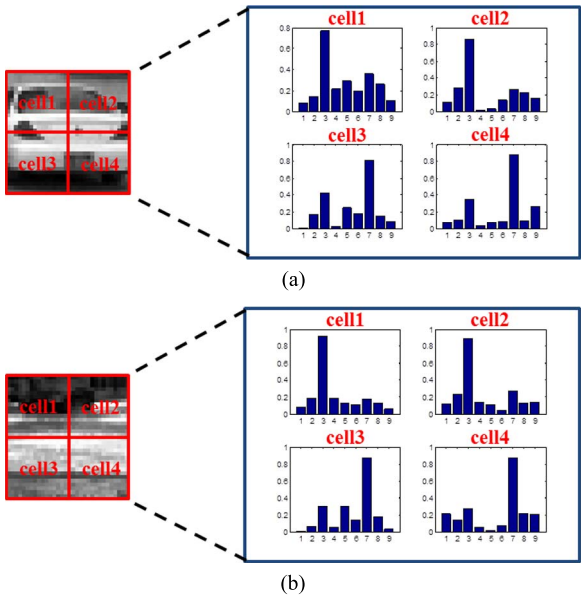


Fig. 5. HOG for a (a) vehicle window and (b) background window.

A. Position Part

As shown in Figs. 4 and 5, two different cells can have the same histogram; consequently, they might also have the same HOG feature. To distinguish two different cells with the same histogram, the positions of the oriented gradients are used as an additional feature in this paper. More specifically, let $\theta(x, y, c)$ be the orientation of the gradient at position (x, y) of the c-th cell and $0 \leq \theta(x, y, c) < 2\pi$. When the orientations of the gradients are quantized into T bins, the orientation bin of the gradient $B(x, y, c)$ is therefore defined as

$$B(x, y, c) = \left\lceil \frac{T\theta(x, y, c)}{2\pi} \right\rceil, \quad (0 \leq \theta(x, y, c) < 2\pi) \qquad (1)$$

where $B(x, y, c) \in \{1, \ldots, T\}$. The means of x and y positions of the d-th bin $(d \in \{1, \ldots, T\})$ in the c-th cell are defined as

$$M^c_{x,d} = \frac{\sum_{x=1}^{c_x} \sum_{y=1}^{c_y} x\mathbb{I}\left[B(x, y, c) = d\right]}{\sum_{x=1}^{c_x} \sum_{y=1}^{c_y} \mathbb{I}\left[B(x, y, c) = d\right]}$$

$$M^c_{y,d} = \frac{\sum_{x=1}^{c_x} \sum_{y=1}^{c_y} y\mathbb{I}\left[B(x, y, c) = d\right]}{\sum_{x=1}^{c_x} \sum_{y=1}^{c_y} \mathbb{I}\left[B(x, y, c) = d\right]} \qquad (2)$$

where c_x and c_y denote cell width and height, respectively; $\mathbb{I}(\cdot)$ is an indicator function that returns a value of one if the argument is true and zero otherwise. Then, the position part of the c-th cell in the new feature is $\mathbf{P}_c = [\mathbf{M}^c_x, \mathbf{M}^c_y]$, where $\mathbf{M}^c_x = [M^c_{x,1}, \ldots, M^c_{x,T}]$ and $\mathbf{M}^c_y = [M^c_{y,1}, \ldots, M^c_{y,T}]$. The computation of the position part from a gradient image is summarized in Fig. 6. In the figure, the orientations of the gradients are divided into nine bins. The first subfigure shows gradients for "cell2," and the second and third subfigures

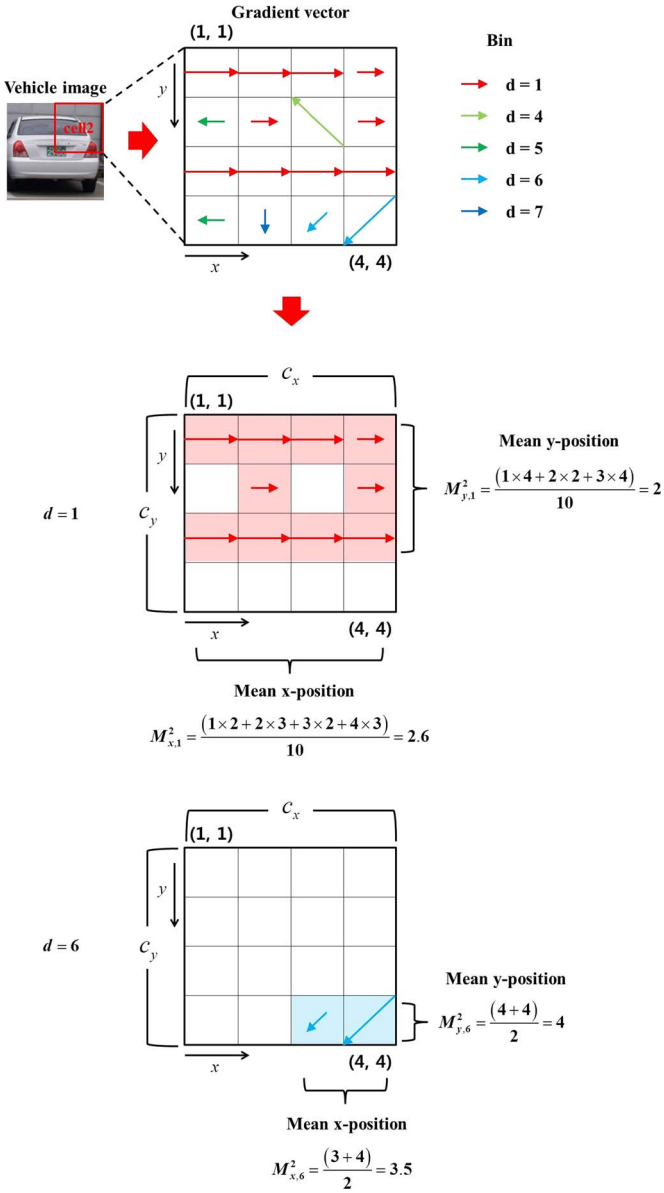


Fig. 6. Procedure for computing the position parts.

explain how $\mathbf{M}^c_x$ and $\mathbf{M}^c_y$ can be computed for $d = 1$ and $d = 6$, respectively. Another example for the position parts is given in Fig. 7, where the two cells have the same HOG but different position parts.

B. Intensity Part

In this subsection, the intensity invariant region (IIR) is defined, and new features are proposed based on the IIR. Consider the set of positive vehicle images

$$\mathcal{V}^+ = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{N_v}\}, \quad \mathbf{s}_i = [s_{i1}, s_{i2}, \ldots, s_{iN}]^T \in \Re^N \qquad (3)$$

where $\mathbf{s}_i$ is the i-th positive vehicle image; s_{ij} is the intensity of the j-th pixel in the i-th vehicle image $\mathbf{s}_i$; N is the size of the image; and N_v is the size of the set of positive images. First, all the vehicle images are normalized to reduce the sensitivity to

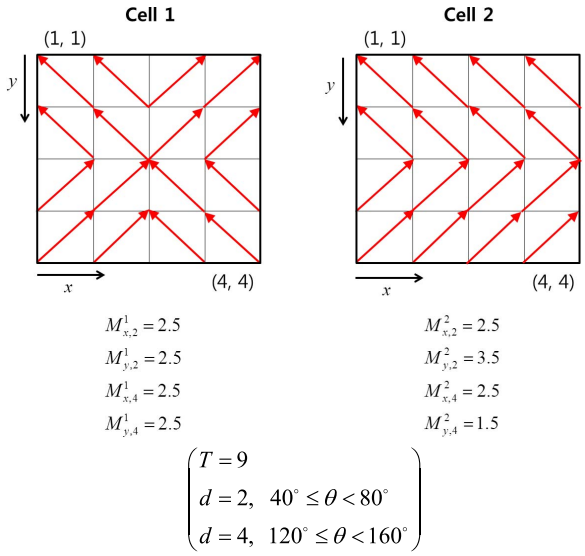


Fig. 7. Two different cells with the same HOG but different position parts.

variation in illumination before computing the intensity part in πHOG. To find the regions in which intensities are invariant across the positive vehicle images, the mean and standard deviation of the vehicle images are respectively computed as

$$\mathbf{M} = \frac{1}{N_v}\sum_{i=1}^{N_v}\mathbf{s}_i = [m_1, m_2, \ldots, m_N]^T \tag{4}$$

$$\boldsymbol{\sigma} = \sqrt{\frac{1}{N_v}\sum_{i=1}^{N_v}(\mathbf{s}_i - \mathbf{M}) \circ (\mathbf{s}_i - \mathbf{M})} = [\sigma_1, \sigma_2, \ldots, \sigma_N]^T \tag{5}$$

where $\circ$ denotes component-wise multiplication. The basic idea of the IIR is that, when the standard deviation image $\boldsymbol{\sigma}$ is computed for positive vehicle images, low standard deviations correspond to the regions (or pixels) in which all training vehicle images have similar intensity values. Thus, intensity values in the regions can be considered unique to the vehicles regardless of the colors or types of vehicles, and the values can be used for vehicle detection. The regions are called IIR. To extract new features from the IIR, we divide the values of $\boldsymbol{\sigma}$ into M intervals and construct the masks $\mathcal{U}_k$ $(k = 1, 2, \ldots, M)$. More specifically, consider the following definitions:

$$\begin{aligned}
\xi_1 &= \text{the 1st smallest in } \boldsymbol{\sigma} = \min\ \boldsymbol{\sigma} \\
\xi_2 &= \text{the } \left\lceil \frac{N}{M} \right\rceil \text{ th smallest in } \boldsymbol{\sigma} \\
\xi_3 &= \text{the } 2 \times \left\lceil \frac{N}{M} \right\rceil \text{ th smallest in } \boldsymbol{\sigma} \\
&\vdots \\
\xi_M &= \text{the } (M-1) \times \left\lceil \frac{N}{M} \right\rceil \text{ th smallest in } \boldsymbol{\sigma} \\
\xi_{M+1} &= \text{the largest in } \boldsymbol{\sigma} = \max\ \boldsymbol{\sigma}.
\end{aligned} \tag{6}$$

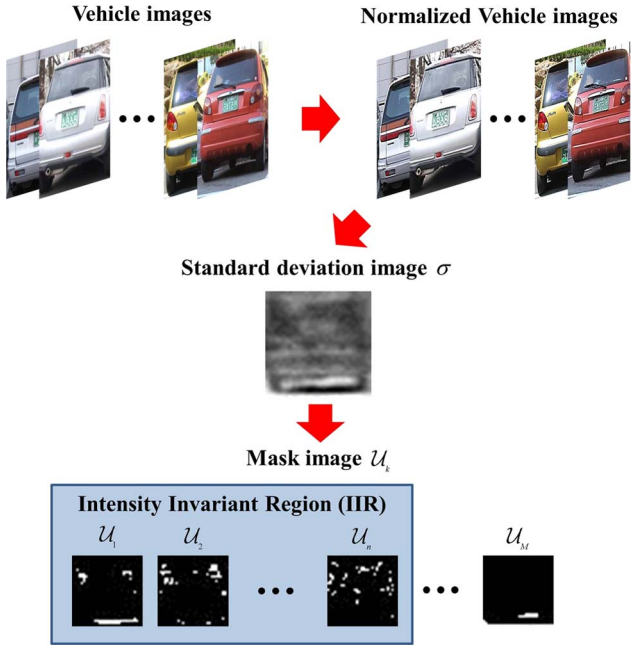


Fig. 8. Example of generation of an IIR.

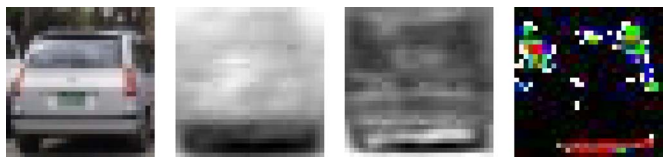


Fig. 9. Results of (a) vehicle image, (b) mean image, (c) standard deviation image, and (d) IIR ($\mathcal{U}_1$: red, $\mathcal{U}_2$: green, $\mathcal{U}_3$: blue, $\mathcal{U}_4$: white).

Then, the kth mask $\mathcal{U}_k$ is defined as a set by applying the two level thresholds to $\boldsymbol{\sigma}$

$$\mathcal{U}_k = \{j | \xi_k \le \sigma_j \le \xi_{k+1}, \quad j = 1, 2, \ldots, N\}. \tag{7}$$

The definition of IIR and some examples are given in Figs. 8 and 9, respectively. In Fig. 9(b), the mean image $\mathbf{M}$ has low intensities (a dark part) in the lower part of the car because cars typically have dark tires and shadows located near the ground.

In Fig. 9(c), the standard deviation image $\boldsymbol{\sigma}$ has small values and appears dark in the regions corresponding to the car outline, tires, and shadow under the car. Fig. 9(d) shows IIR $\mathcal{U}_k$ when $M = 20$. Only the first $n = 4$ masks are considered, and $\mathcal{U}_1$, $\mathcal{U}_2$, $\mathcal{U}_3$, and $\mathcal{U}_4$ are indicated in red, green, blue, and white, respectively. Then, assume that a test image $\mathbf{s} \in \Re^N$ is presented. A standard normal deviate image is computed as

$$\mathbf{z} = \left(\frac{1}{\boldsymbol{\sigma}}\right) \circ (\mathbf{s} - \mathbf{M}) \in \Re^N. \tag{8}$$

The intensities of $\mathbf{z}$ correspond to the IIR masks according to

$$h_k = \frac{1}{|\mathcal{U}_k|}\sum_{j \in \mathcal{U}_k} z_j. \tag{9}$$

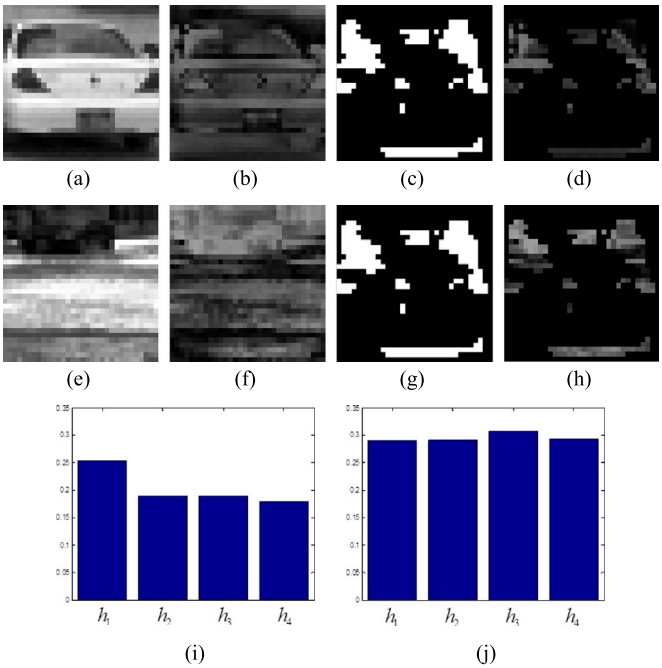


Fig. 10. Procedure for extracting the intensity part of πHOG according to vehicle and background images.

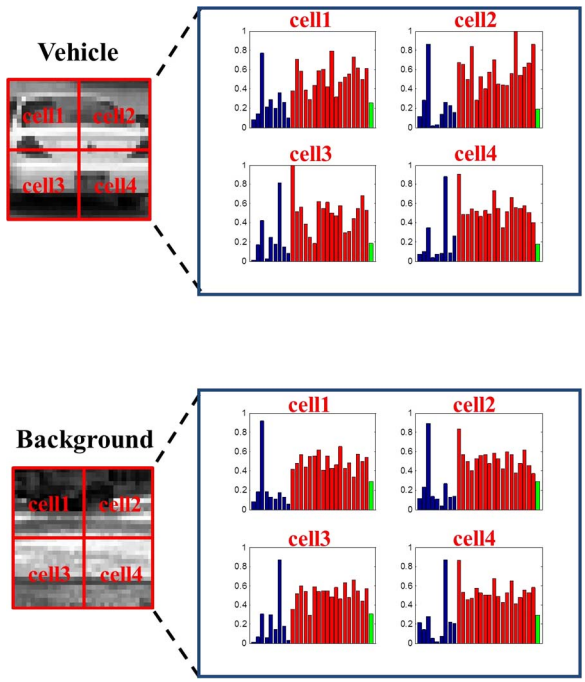


Fig. 11. πHOG features are denoted according to the vehicle window and background window ($T = 9,\ C = 4,\ n = 4$).

Here, $\mathcal{U}_k$ are used as new features, where $\mathbf{z} = [z_1, z_2, \ldots, z_N]^T$ and $k = 1, \ldots, n$. Fig. 10 explains how the features from the IIR are computed when a new image $\mathbf{s}$ is presented. Here, $M = 20$ and $n = 4$. Fig. 10(a) and (b) show a new *vehicle* image $\mathbf{s}_{veh}$ and the associated standard normal deviate image $|\mathbf{z}_{veh}|$, respectively. Fig. 10(c) and (d) are IIR masks $(\bigcup_{k=1}^{n} \mathcal{U}_k)$ and associated IIR features $h_{veh,k} = (1/|\mathcal{U}_k|) \sum_{j \in \mathcal{U}_k} z_{veh,j}$ $(k = 1, \ldots, n)$ for vehicle image $\mathbf{s}_{veh}$, respectively. The area of IIR masks $\bigcup_{k=1}^{n} \mathcal{U}_k$ covers approximately 20% of the test image. Fig. 10(e)–(h) are the results when the test image is not a vehicle but contains only background. That is, Fig. 10(e) is background image $\mathbf{s}_{bgr}$ and Fig. 10(f)–(h) are the associated $|\mathbf{z}_{bgr}|$, $\bigcup_{k=1}^{n} \mathcal{U}_k$, and $h_{bgr,k} = (1/|\mathcal{U}_k|) \sum_{j \in \mathcal{U}_k} z_{bgr,j}$ $(k = 1, \ldots, n)$, respectively. Fig. 10(i) and (j) are IIR features $h_k = (1/|\mathcal{U}_k|) \sum_{j \in \mathcal{U}_k} z_j$ $(k = 1, \ldots, n)$ for the test images in Fig. 10(a) and (e), respectively. It is worth noting that $h_{veh,k}$ has a lower value than the corresponding $h_{bgr,k}$, and the IIR features have different patterns for the two different images; therefore, these features are important clues in distinguishing vehicles from the background.

C. πHOG

By combining position and intensity parts, πHOG is realized. The πHOG process starts like the conventional HOG. A single window is decomposed into C cells, and the orientations of the gradients in each cell are assigned to T bins. Thus, the HOG of a window has CT dimensions (C cells $\times$ T bins). The position part of πHOG in c-th cell $\mathbf{P}_c$ is represented by a $2T$-dimensional vector (x and y positions for each bin). Then, the position part of πHOG in a window has $2CT$ dimensions. The intensity part of πHOG in a window is the IIR feature h_k $(k = 1, \ldots, n)$ and has n dimensions. Finally, πHOG is defined as

$$F_{\pi\text{HOG}} = [F_{\text{HOG}}, \mathbf{P}_1, \ldots, \mathbf{P}_C, h_1, \ldots, h_n] \qquad (10)$$

where F_{HOG} denotes the HOG feature. Thus, πHOG has $3CT + n$ dimensions. Fig. 11 shows the πHOGs of a vehicle window and a background window. The vehicle and background windows in Fig. 11 are the same as those in Fig. 5. In Fig. 11, the blue bar corresponds to HOG F_{HOG}, and the red and green bars correspond to position part $(\mathbf{P}_1, \ldots, \mathbf{P}_C)$ and intensity part $(h_1, \ldots, h_n)$ of πHOG, respectively. The figure shows that the F_{HOG}s are similar for both windows, while the position and intensity parts of πHOG have different values and can be used to distinguish the two windows. Thus, πHOG has more discriminative power than the conventional HOG due to the additional position and intensity parts. In the following sections, πHOG is used with a support vector machine (SVM) for vehicle detection.

IV. Search Space Reduction

An exhaustive search by sliding a window is a simple yet popular method for vehicle detection in an image. The method scans the whole image while changing the position and size (scale) of the window, as shown in Fig. 12(a). Denote a window as a three-dimensional vector $\mathbf{w} = (w_x \quad w_y \quad w_s)^T$, where $(w_x,\ w_y)$ and w_s are the center position and the size of the window, respectively, as shown in Fig. 12(b). The exhaustive search incurs a high computational cost and requires a large amount of time. In particular, because the proposed πHOG is longer than the conventional HOG, the simple exhaustive search is not a good choice. In this paper, a new search area

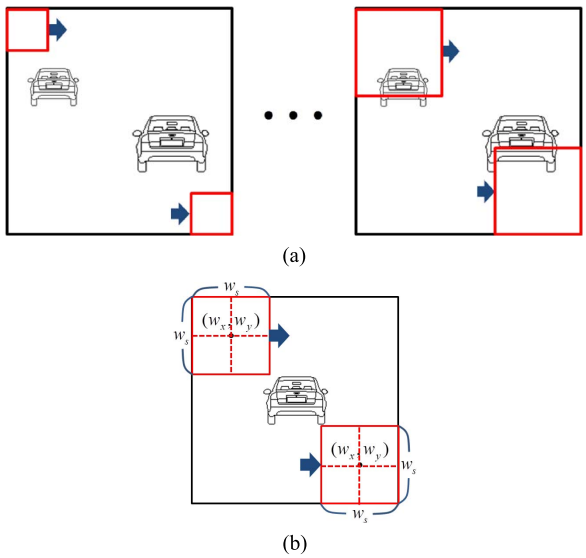


Fig. 12. (a) Exhaustive search by sliding a window, and (b) the window is denoted as a three-dimensional vector in the image.

Fig. 13. Example of high correlation with vehicle size and position in a road environment.

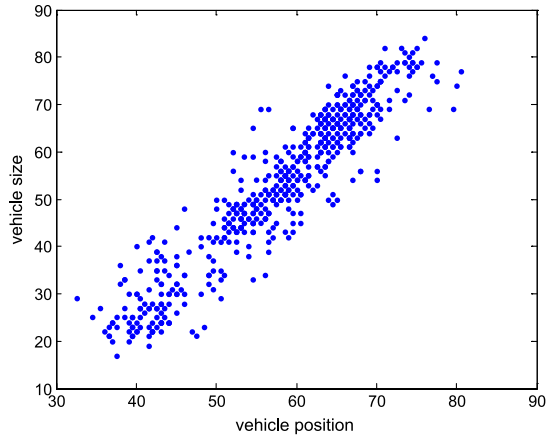


Fig. 14. Vehicle sizes with respect to vehicle positions.

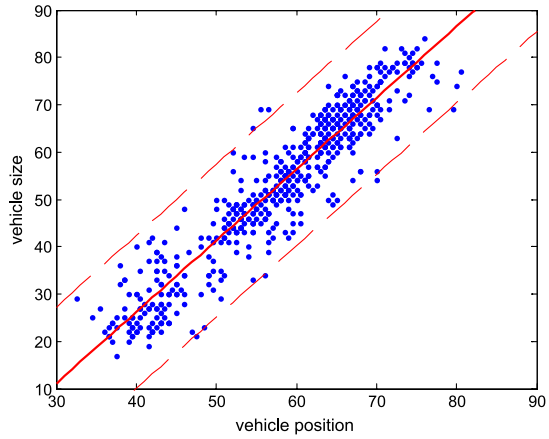


Fig. 15. Position-wise vehicle size predictor in the Caltech dataset.

reduction method is proposed to reduce the computational cost and speed up the detection. The main idea is taken from our previous work [24]. *In that paper, it was reported that the vehicle position in the y-axis is highly correlated with the vehicle size; therefore, we are not required to try all possible combinations of w_y and w_s.* More specifically, large vehicles are detected in the lower part of the image, while small vehicles are detected in the upper part of the image, as shown in Fig. 13.

This idea was confirmed using 500 vehicle windows in the Caltech dataset [43]. The vehicle position in the y-axis is plotted against vehicle size in Fig. 14, where each dot corresponds to a vehicle in the dataset. As stated, small vehicle windows tend to be detected in the upper part of the image (with low y values), while large vehicle windows tend to be detected in the lower part of the image (with high y values). Using this idea, we narrow the search space in $(w_x,\ w_y,\ w_s)$ instead of employing the full and exhaustive search over the whole search space. The SSR method proposed in this paper is outlined as follows.

1) The space of $(w_x,\ w_y,\ w_s)$, in which the vehicles are likely to be detected, is modeled as the linear statistical model

$$w_s = \beta_0 + \beta_1 w_y + \varepsilon$$
$$\varepsilon \sim \mathcal{N}\left(0, \frac{1}{\phi}\right) \tag{11}$$

where $\mathcal{N}(\cdot)$ is a Gaussian distribution, and ϕ denotes precision (the inverse of the variance). Equation (11) is called a position-wise vehicle size predictor (PVSP) and is depicted in Fig. 15.

2) The SSR method is conducted using πHOG and SVM with initial values of $\boldsymbol{\beta} = (\beta_0 \quad \beta_1)^T$ and ϕ.
3) When an image is presented and a vehicle is detected, we obtain a pair $(w_y^1,\ w_s^1)$ and use them to update $\boldsymbol{\beta}$ and ϕ of the PVSP. Here, the superscript "1" in $(w_y^1,\ w_s^1)$ denotes the first detected vehicle. Hereafter, the superscript n in $\mathbf{w}$, such as $\mathbf{w}^n = (w_x^n \quad w_y^n \quad w_s^n)^T$, refers to the n-th detected vehicle.
4) When another image is presented, we narrow the search space using the PVSP with the updated parameters $\boldsymbol{\beta}$ and ϕ. That is, we restrict the size of the window into k standard deviations from the mean $\beta_0 + \beta_1 w_y$ using

$$w_s \in \left[\beta_0 + \beta_1 w_y - \frac{k}{\sqrt{\phi}}, \quad \beta_0 + \beta_1 w_y + \frac{k}{\sqrt{\phi}}\right]$$
$$= \left[\mathbf{w}_y^T \boldsymbol{\beta} - \frac{k}{\sqrt{\phi}}, \quad \mathbf{w}_y^T \boldsymbol{\beta} + \frac{k}{\sqrt{\phi}}\right] \tag{12}$$

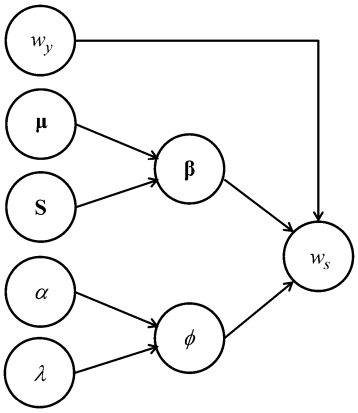


Fig. 16. Graphical model of the PVSP.

where $\boldsymbol{\beta} = (\beta_0 \quad \beta_1)^T$, and $\mathbf{w}_y = (1 \quad w_y)^T$. If another vehicle is detected, the above steps are repeated, and $\boldsymbol{\beta}$ and ϕ are updated. In this paper, we choose $k = 3$ because three standard deviations from the mean account for 99.7% of the samples. For systematic and efficient update of parameters $\boldsymbol{\beta}$ and ϕ of the PVSP, we use a Bayesian approach. We assume that $\boldsymbol{\beta}$ and ϕ are random variables, and that the PVSP is represented by the graphical model in Fig. 16.

In subsequent developments, denote the n-th detected vehicle as $\mathbf{w}^n = (w_x^n \quad w_y^n \quad w_s^n)^T$ and the accumulated vehicles as $\mathbf{W}^n = \{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^n\}$. Denote the parameters of the PVSP after detecting n vehicles as $(\boldsymbol{\beta}, \phi)|\mathbf{W}^n$. If we assume

$$(\boldsymbol{\beta}, \phi)|\mathbf{W}^{n-1} \sim \mathcal{NG}(\boldsymbol{\mu}^{n-1}, \mathbf{S}^{n-1}, \alpha^{n-1}, \lambda^{n-1}) \tag{13}$$

and that the n-th vehicle $\mathbf{w}^n = (w_x^n \quad w_y^n \quad w_s^n)^T$, which respects the PVSP,

$$w_s^n \sim \mathcal{N}\left(\cdot|\beta_0 + \beta_1 w_y^n, \frac{1}{\phi}\right) \tag{14}$$

is detected, then the parameters of the PVSP can be updated using the Bayesian rule as

$$(\boldsymbol{\beta}, \phi)|\mathbf{W}^n \sim \mathcal{NG}(\cdot|\boldsymbol{\mu}^n, \mathbf{S}^n, \alpha^n, \lambda^n) \tag{15}$$

$$\begin{aligned}
&\alpha^n = \alpha^{n-1} + \frac{1}{2},\ (\mathbf{S}^n)^{-1} = \left\{(\mathbf{S}^{n-1})^{-1} + \mathbf{w}_y^n \left(\mathbf{w}_y^n\right)^T\right\} \\
&\boldsymbol{\mu}^n = \mathbf{S}^n \left\{(\mathbf{S}^{n-1})^{-1}\boldsymbol{\mu}^{n-1} + \mathbf{w}_y^n w_s^n\right\} \\
&\lambda^n = \lambda^{n-1} + \frac{1}{2}\left\{(\boldsymbol{\mu}^{n-1})^T(\mathbf{S}^{n-1})^{-1}\boldsymbol{\mu}^{n-1} + (w_s^n)^2 \right. \\
&\qquad\qquad \left. - (\boldsymbol{\mu}^n)^T(\mathbf{S}^n)^{-1}(\boldsymbol{\mu}^n)\right\}
\end{aligned} \tag{16}$$

where $\mathcal{N}(\cdot)$ and $\mathcal{NG}(\cdot)$ denote the normal and normal-gamma distributions, respectively. Here, the normal-gamma distribution $\mathcal{NG}(\cdot)$ is defined as

$$\begin{aligned}
&\mathcal{NG}(\boldsymbol{\beta}, \phi|\boldsymbol{\mu}, \mathbf{S}, \alpha, \lambda) \\
&\quad = \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}, \phi^{-1}\mathbf{S})\mathcal{G}(\phi|\alpha, \lambda) \\
&\quad = \frac{1}{\sqrt{\det(2\pi\phi^{-1}\mathbf{S})}} \exp\left\{-\frac{\phi}{2}(\boldsymbol{\beta} - \boldsymbol{\mu})^T\mathbf{S}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu})\right\} \\
&\qquad \times \frac{1}{\Gamma(\alpha)}\lambda^\alpha \phi^{\alpha-1}\exp(-\lambda\phi) \\
&\qquad \propto \phi^\alpha \exp\left[-\frac{\phi}{2}\left\{2\lambda + (\boldsymbol{\beta} - \boldsymbol{\mu})^T\mathbf{S}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu})\right\}\right]
\end{aligned} \tag{17}$$

where $\mathcal{G}(\cdot)$ and $\Gamma(\alpha)$ denote the gamma distribution and gamma function, respectively. The derivation of (16) is similar to the one given in [41], [49], except that precision ϕ is considered to be unknown, and $(\boldsymbol{\beta}, \phi)$ is modeled as a normal-gamma distribution. The derivation is summarized in the Appendix. Using the PVSP parameterized by $(\boldsymbol{\beta}, \phi)|\mathbf{W}^n$, we refine the search space. If we replace $\boldsymbol{\beta}$ and ϕ in (12) with their means $\mathbb{E}(\boldsymbol{\beta}|\mathbf{W}^n)$ and $\mathbb{E}(\phi|\mathbf{W}^n)$, respectively, then we obtain

$$\begin{aligned}
w_s \in \Bigg[&\mathbf{w}_y^T\mathbb{E}(\boldsymbol{\beta}|\mathbf{W}^n) - \frac{k}{\sqrt{\mathbb{E}(\phi|\mathbf{W}^n)}}, \\
&\mathbf{w}_y^T\mathbb{E}(\boldsymbol{\beta}|\mathbf{W}^n) + \frac{k}{\sqrt{\mathbb{E}(\phi|\mathbf{W}^n)}}\Bigg].
\end{aligned} \tag{18}$$

From [42],

$$\mathbb{E}(\boldsymbol{\beta}|\mathbf{W}^n) = \boldsymbol{\mu}^n, \quad \mathbb{E}(\phi|\mathbf{W}^n) = \frac{\alpha^n}{\lambda^n}. \tag{19}$$

For the normal-gamma distribution $\mathcal{NG}(\cdot)$. Then, the proposed SSR method is summarized as in Table I.

The body of the first loop in Table I is applied to all images. In Line 2 of Table I, the parameters of the PVSP, $\boldsymbol{\beta}^n, \phi^n$, are estimated using (19) and the hyper-parameters $\boldsymbol{\mu}^n, \mathbf{S}^n, \alpha^n, \lambda^n$. In lines 4 to 7 in Table I, sliding windows are used within the PVSP. In lines 8 through 10, πHOG + SVM is applied to the windows. If a vehicle is detected, parameters $\boldsymbol{\mu}^n, \mathbf{S}^n, \alpha^n, \lambda^n$ are updated using (16).

V. Experimental Results

In this section, the proposed method is tested with four datasets: Caltech [43], IR, Pittsburgh [44], and Kitti sets [45]. Each dataset is built using a fixed monocular camera installed on the front part of the host vehicle. Then, the proposed method is compared with a variety of combinations of previous features, HGs/SSRs, and classifiers. As previous features, standard HOG, local binary pattern (LBP) [46], local gradient pattern (LGP) [31], and aggregate channel feature (ACF) [20] are used. As HG or SSR, the exhaustive search (ES), shadow (SHDW) [9], horizontal/vertical edges (VH) [11], and optical flow (OF) [15] methods are compared with the proposed method (PVSP). As classifiers, SVM [34], ELM [35], and kNN [36] are used. The comparison is conducted in terms of detection performance

TABLE I
PROPOSED SSR METHOD

Initialize the hyper-parameter

$\boldsymbol{\mu}^0 = [-100 \quad 2]$, $\mathbf{S}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\alpha^0 = 1$, $\lambda^0 = 1$, $n = 0$

1: for each image
2: estimate $\boldsymbol{\beta}^n, \phi^n$ from $\boldsymbol{\mu}^n, \mathbf{S}^n, \alpha^n, \lambda^n$
3: $\boldsymbol{\beta}^n = \mathbb{E}(\boldsymbol{\beta}^n \mid \mathbf{W}^n) = \boldsymbol{\mu}^n$, $\phi^n = \mathbb{E}(\phi^n \mid \mathbf{W}^n) = \dfrac{\alpha^n}{\lambda^n}$
4: for $w_x = 1$ to width
5: for $w_y = 1$ to height
6: $\mathbf{w}_y = \begin{pmatrix} 1 & w_y \end{pmatrix}^T$
7: for $w_s = \mathbf{w}_y^T \boldsymbol{\beta}^n - \dfrac{k}{\sqrt{\phi^n}}$ to $\mathbf{w}_y^T \boldsymbol{\beta}^n + \dfrac{k}{\sqrt{\phi^n}}$
8: if the vehicle is detected (πHOG-SVM $(w_x, w_y, w_s) > 0$)
9: $n = n + 1$
10: $w_x^n = w_x$, $w_y^n = w_y$, $w_s^n = w_s$
11: update parameter $\boldsymbol{\mu}^n, \mathbf{S}^n, \alpha^n, \lambda^n$
12: $(\mathbf{S}^n)^{-1} = \left\{ (\mathbf{S}^{n-1})^{-1} + \mathbf{w}_y^n (\mathbf{w}_y^n)^T \right\}$
13: $\boldsymbol{\mu}^n = \mathbf{S}^n \left\{ (\mathbf{S}^{n-1})^{-1} \boldsymbol{\mu}^{n-1} + \mathbf{w}_y^n w_s^n \right\}$
14: $\lambda^n = \lambda^{n-1} + \dfrac{1}{2} \left\{ (\boldsymbol{\mu}^{n-1})^T (\mathbf{S}^{n-1})^{-1} \boldsymbol{\mu}^{n-1} + (w_s^n)^2 - (\boldsymbol{\mu}^n)^T (\mathbf{S}^n)^{-1} (\boldsymbol{\mu}^n) \right\}$
15: $\alpha^n = \alpha^{n-1} + \dfrac{1}{2}$
16: end
17: end
18: end
19: end
20: end

and computation time. In subsequent experiments, the PASCAL measure [47] is used to evaluate detection performance. Thus, if the overlap r_n between the ground truth and the n-th detected vehicle defined by

$$r_n = \frac{area(GT \cap \mathbf{w}^n)}{area(GT \cup \mathbf{w}^n)} \tag{20}$$

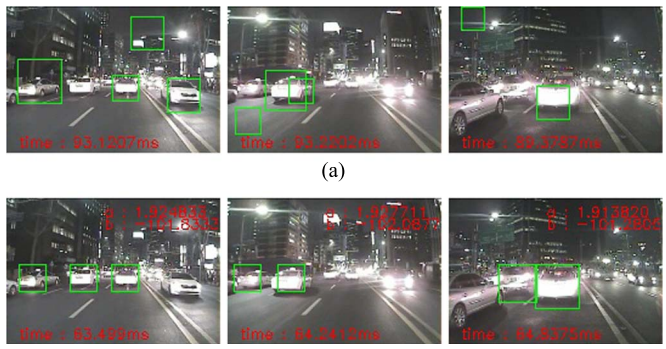
exceeds a threshold T_n, we consider our detection to be correct; otherwise, the detection is wrong, where GT is the ground truth. In this paper, $T_n = 0.6$ is used.

A. Caltech Dataset

The Caltech dataset consists of 464 images and 588 ground truths, which include the rear portions of vehicles. The size of each image is 180×120 pixels. Because the Caltech set is relatively small, the set is used only as a test set. Another set from [48] is used as a training set to extract the features and train the classifiers. The number of training images taken from [48] is 1988. The SVM is trained using 2,903 positive windows and 2,991 negative windows collected from 1988 training images. The training window is resized to 32×32. Fig. 17 shows some vehicle detection results for the Caltech dataset.

(a) (b)

Fig. 17. Detection results using the Caltech dataset: (a) previous method (HOG + ES + SVM) and (b) proposed method (πHOG + PVSP + SVM) (frames 110, 131, 390).



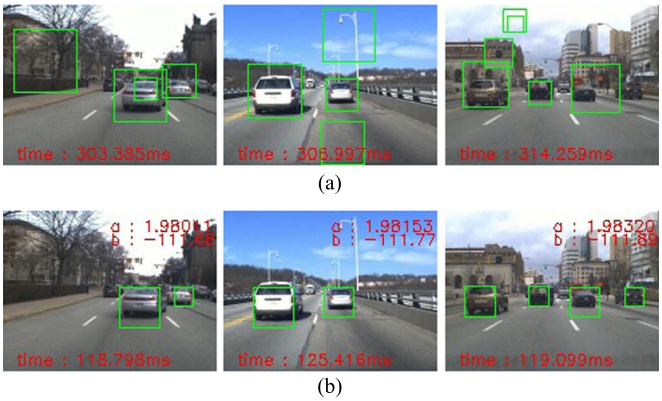
(a) (b)

Fig. 18. Detection results using the IR dataset: (a) previous method (HOG + ES + SVM) and (b) proposed method (πHOG + PVSP + SVM) (frames 39, 62, 197).

The first row was achieved using HOG + ES, and the second row was achieved using the proposed method (πHOG + PVSP). For both methods, a linear SVM classifier was employed for fast evaluation. As shown in the figure, the proposed method had fewer false negatives (frame 110, 131, 390) and required less time than the previous method. In Fig. 17(b), a and b denote PVSP parameters β_1 and β_0, respectively.

B. IR Dataset

The images in the IR dataset were captured using a near-infrared (NIR) camera. The database includes 204 test images and 955 training images. The test images include 238 ground truths. The size of the images was 176×120. A linear SVM was trained using 1,144 positive windows and 2,416 negative windows from the 955 training images. The training image was resized to 32×32. The detection results of the previous and proposed methods are compared for frames 39, 62, and 197 in Fig. 18. In the figure, the first row corresponds to the previous method (HOG + ES), while the second row corresponds to the proposed method (πHOG + PVSP). The proposed method demonstrates fewer false positives and false negatives than the previous method. In addition, the proposed method takes less time than the previous one.



(a) (b)

Fig. 19. Detection results from the Pittsburgh dataset: (a) previous method (HOG + ES + SVM) and (b) proposed method (πHOG + PVSP + SVM) (frames 98, 274, 476).

(a) (b)

Fig. 20. Detection results in the Kitti dataset: (a) previous method (HOG + ES + SVM) and (b) proposed method (πHOG + PVSP + SVM) (frames 50, 122, 214).

C. Pittsburgh Dataset

The Pittsburgh dataset is comprised of 6,109 images and 14,453 ground truths collected in various locations and under various weather conditions. In this dataset, there are four kinds of vehicles: sedans, SUVs, trucks, and buses. In this experiment, we focused on the sedans and SUVs because of their similarity in size. From 6,109 images, 4,161 and 600 were selected randomly as training and test images, respectively. The test images include 1330 ground truths. A linear SVM was trained using 6,143 positive windows and 7,118 negative windows clipped from the 4,161 training images. All training windows were resized to 32×32, and the test images were resized to 160×120. The detection results of the previous (HOG + ES) and proposed (πHOG + PVSP) methods are compared in Fig. 19. The results are those of frames 98, 274, and 476.

As in the previous two datasets, the proposed method demonstrated fewer false positives and false negatives than the previous method. This dataset contains various vehicle sizes. The Pittsburgh dataset has more small cars than the previous two datasets; nevertheless, the proposed method still demonstrated consistently less FP and FNs.

D. Kitti Dataset

The Kitti dataset consists of 7481 training images and 7518 test images. From 7481 training images, 6481 and 1000 were selected randomly as training and test images, respectively, for the sake of simplicity. The test images include 743 ground truths. A linear SVM was trained using 6,983 positive windows and 7,118 negative windows clipped from the 6481 training images. All training windows were resized to 32×32, and the test images were resized to 397×120. In this dataset, various vehicles are captured from various angles and overlapped one other. The detection results of the previous (HOG + ES) and proposed (πHOG + PVSP) methods are compared in Fig. 20. The results are those of frames 98, 274, and 476.

The Kitti dataset includes a larger number of complex images than the previous three datasets, but the proposed method still demonstrates fewer FP and FN compared with other methods.

TABLE II
SSRs, HGs, Features, and Classifiers

HGs/ SSRs	Features	Classifiers
SHADOW	HOG	
VH	LBP	SVM
OP	LGP	ELM
ES	ACF	KNN
PVSP	πHOG	

E. Analysis of the Experimental Results

In this subsection, a variety of combinations of HGs/SSRs, features, classifiers, and training sets are tested and compared with the other methods in terms of detection error tradeoff (DET) curves. The DET curve consists of false positives per image (FPPI) and a false rejection rate (FRR); the goal of vehicle detection is to simultaneously decrease both the FAR and FRR and drive the DET curve to the lower left corner. The HGs/SSRs, features, and classifiers used in the experiments are summarized in Table II.

First, a linear SVM is used, and the experimental results are summarized in Fig. 21. Concerning the features extracted from a window, the proposed πHOG is compared with the standard HOG, LBP, LGP and ACF. As shown in Fig. 21, the πHOG outperforms the other features in all four datasets when using the same SSR. Concerning the SSR, the PVSP is compared with the ES. In the figures, the dotted lines correspond to the PVSP, while the solid lines correspond to the ES. Fig. 21 shows *that the proposed PVSP demonstrates lower FRR and FPPI than does the ES for all tested features*. Initially, the PVSP was intended to speed up the detection; however, it also lowered the FPPI by focusing on the space in which the vehicles are likely to be detected.

In Fig. 22, the PVSP is compared with other HGs such as SHDW, VH, and OF. In the figure, the dotted lines correspond to the πHOG, while the solid lines correspond to the HOG. The PVSP shows the best performance among other SSRs and HGs in all three datasets. In particular, SHDW demonstrates degraded performance in IR and Pittsburgh datasets. VH also demonstrates degraded performance in IR, Pittsburgh, and Kitti datasets. The reasons for the performance degradation of

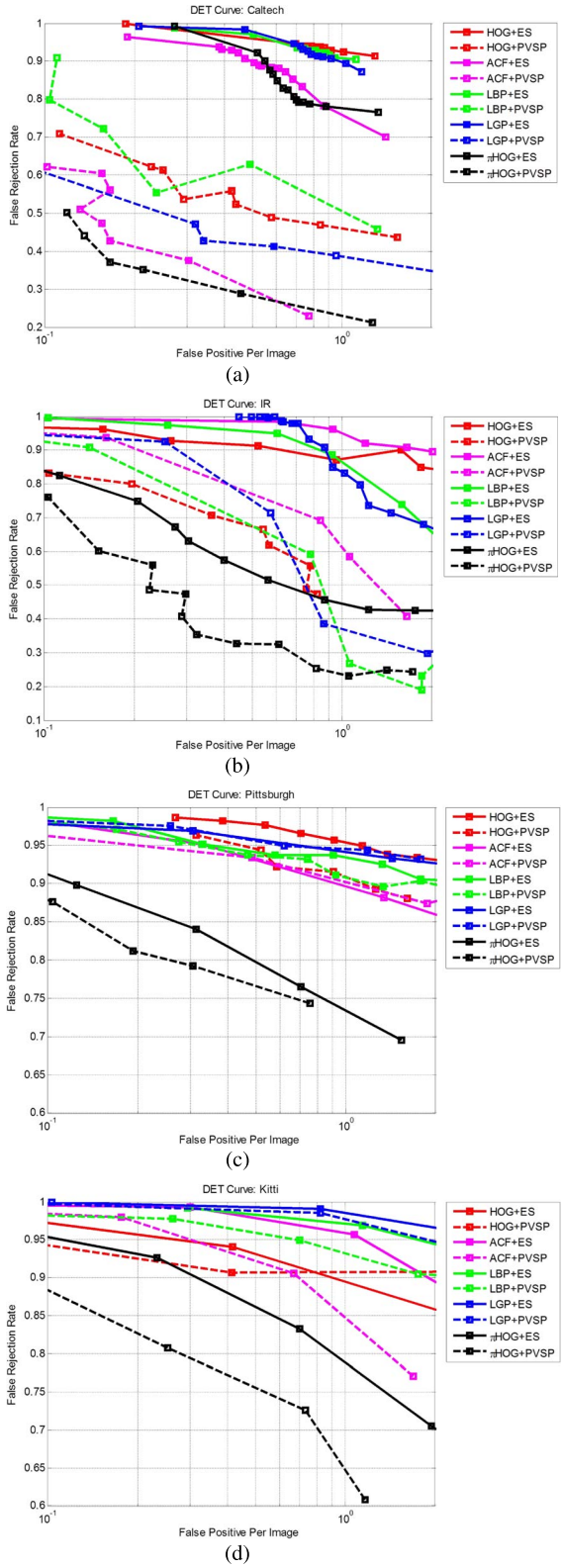


Fig. 21. DET curves using different features for the (a) Caltech dataset, (b) IR dataset, (c) Pittsburgh dataset, and (d) Kitti dataset (ES/PVSP as SSRs and SVM as a classifier).

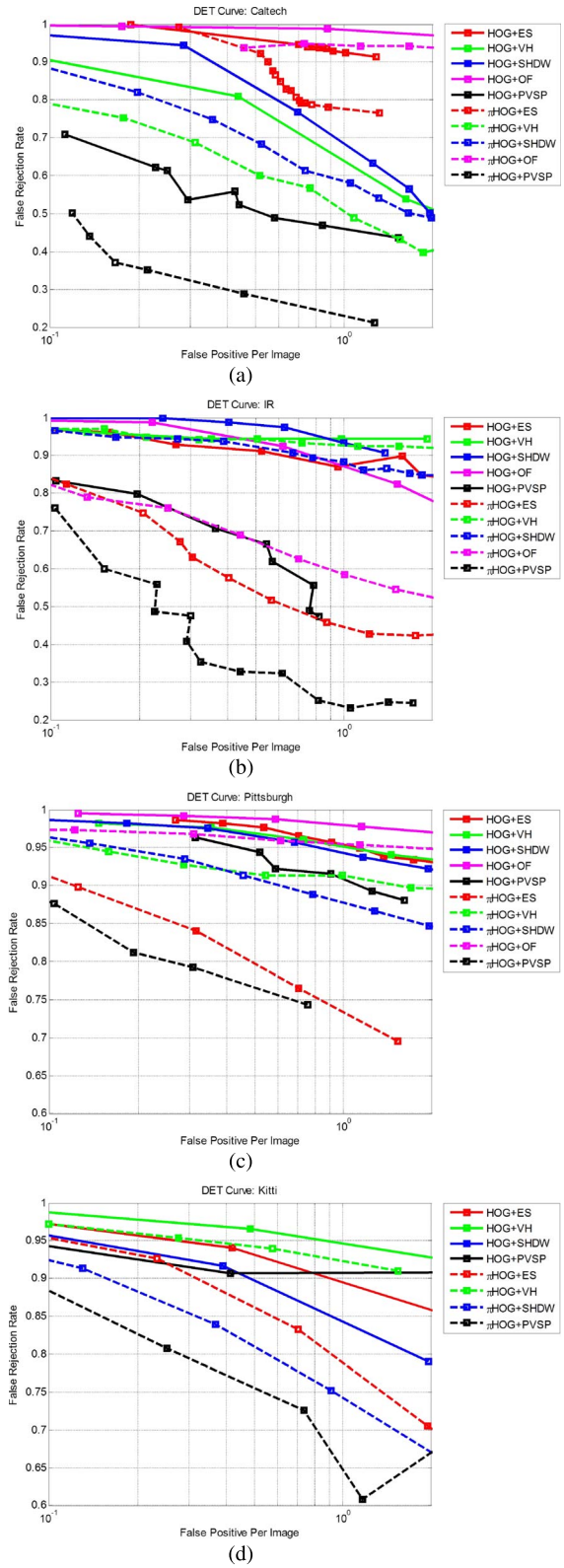


Fig. 22. DET curves using different SSRs and HGs for the (a) Caltech dataset, (b) IR dataset, (c) Pittsburgh dataset, and (d) Kitti dataset (HOG/πHOG as features and SVM as a classifier).

SHDW and VH in the above datasets are that 1) in the IR dataset, shadows and horizontal/vertical edges are frequently missed because all the images were collected at night; 2) in the Pittsburgh dataset, VH and SHDW often identify erroneous windows because the images in the dataset include a number of vehicles with different sizes; 3) in the Kitti dataset, some

vehicle images are taken from various angles so that some of the vertical edges are not detected, resulting in VH erroneously identifying vehicles. The OF shows the worst performance among the HGs/SSRs because the test images in the datasets are still images, and the difference between two consecutive images is too large to be properly addressed using OF. In particular, the test images in the Kitti dataset are all still images; thus, OF cannot be evaluated in the Kitti dataset.

Using different classifiers, similar experiments were conducted. In Fig. 23, the SVM is compared with the ELM and kNN. For the sake of simplicity, only PVSP is used as a SSR.

Overall, the SVM outperforms the ELM and kNN in Caltech, Pittsburgh, and Kitti datasets, while the ELM and kNN show better performance than the SVM in the IR set. Considering only the DET curve, the performances of the SVM, ELM, and kNN are similar, and no conclusion can be drawn about the preference for classifier.

A larger training set is then used to observe how the training set affects the detection performance. In this experiment, the union of the four test sets used in Figs. 21–23 is employed as a single training set. The experimental result is summarized in Fig. 24. In the legends of the figures, "CAL," "IR," "PT," and "KIT" indicate that the set in [48] and the IR, Pittsburg, and Kitti sets are collectively used as a training set. "TOTAL" in the legend means that the three datasets "CAL," "IR," and "PT" are collectively used as a training set. Thus, a total of 10,190 positive windows and 12,524 negative windows are used for "TOTAL." In these figures, the dotted lines correspond to the "TOTAL," while the solid lines correspond to the corresponding single training set. For the sake of simplicity, a linear SVM was used to detect the vehicles. In addition, position and intensity parts of πHOG were also compared separately with HOG and πHOG to determine the effects of both individual parts. PHOG and IHOG are the position and intensity parts of πHOG, respectively. Overall, the influence of the use of the "TOTAL" dataset on the performance was mixed. In some cases, the use of the "TOTAL" dataset improved the performance; however, in other cases, it had adverse effects on the performance. Thus, no clear conclusion could be drawn about the use of a "TOTAL" training set. The reason for the adverse effect of the "TOTAL" training set might be that the additional training windows are different from the test samples and degrade the detection performance. Instead, it was *consistently* observed that both PHOG and IHOG improved the performance compared to that of HOG, contributing separately to πHOG. Generally, it was concluded that the position part makes a larger contribution to πHOG than does the intensity part, and that πHOG $>$ PHOG $>$ IHOG $>$ HOG with regard to impact on performance.

Finally, the bus and truck images were included in a training set in order to observe how the detection performance was changed when the vehicles had different sizes. This experiment was conducted only for Pittsburg dataset because the set has separate "SEDAN," "SUV," "BUS," and "TRUCK" sets.

In Fig. 25, "SS" means that only sedan and SUV images are used as a training set, while "SSBT" means that bus and truck images are also included in a training set along with sedan and SUV images. A linear SVM was also trained using 6,928 positive windows and 7,118 negative windows. It is interesting that the inclusion of "BUS" and "TRUCK" in the training set degrades the performances of PHOG, IHOG and πHOG but not

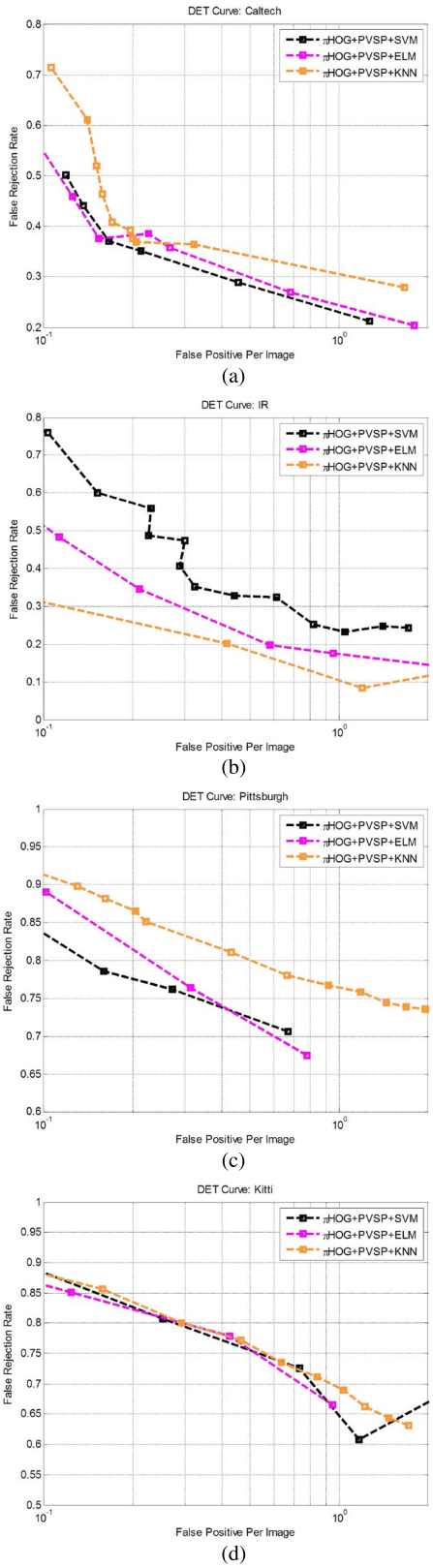


Fig. 23. DET curves using different classifiers for the (a) Caltech dataset, (b) IR dataset, (c) Pittsburgh dataset, and (d) Kitti dataset (πHOG as a feature and PVSP as a SSR).

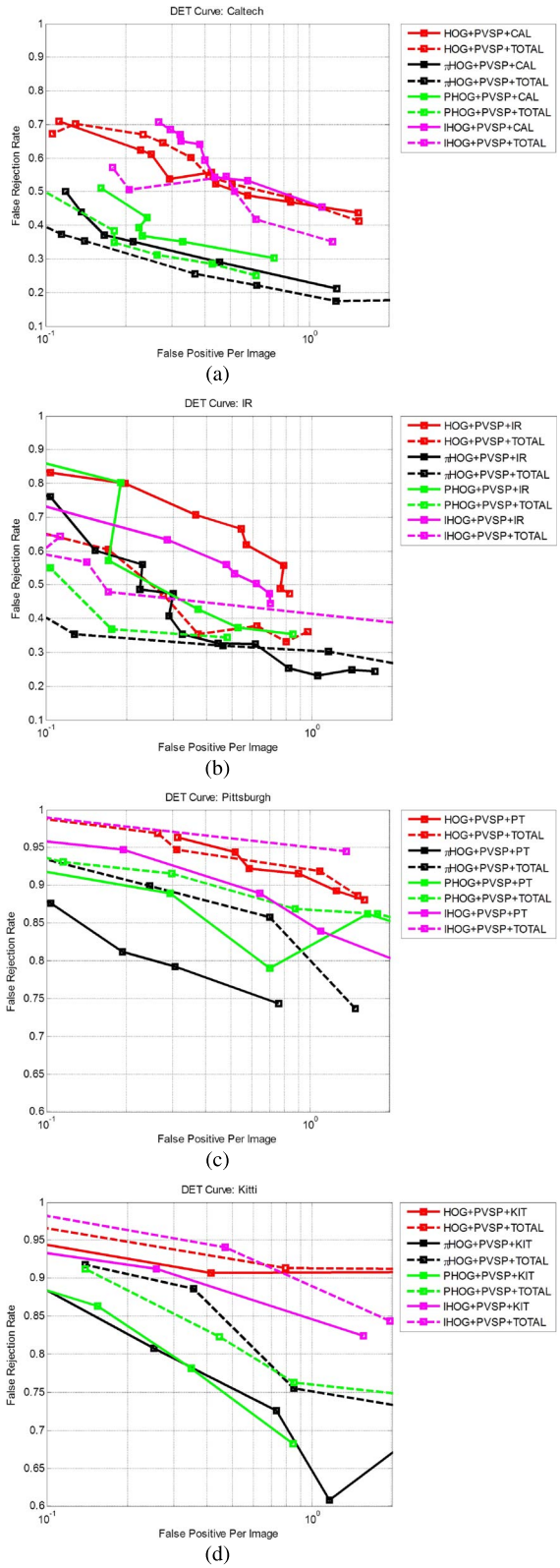


Fig. 24. DET curves using different training sets for the (a) Caltech dataset, (b) IR dataset, (c) Pittsburgh dataset, and (d) Kitti dataset (SVM as a classifier, HOG/πHOG as features, and PVSP as SSRs).

that of HOG. The reason for the degradation in πHOG could be that the IIR pixels in πHOG are affected by the different sizes of the vehicles. The position and intensity parts, however, are still

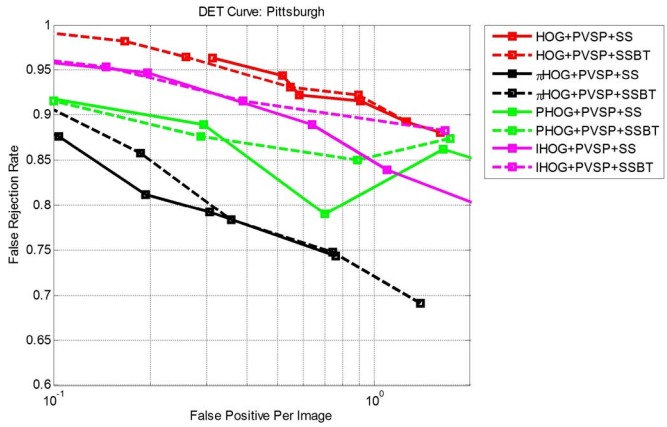


Fig. 25. DET curves using different training sets for the Pittsburgh dataset (SVM as a classifier, HOG/πHOG as features, and ES/PVSP as SSRs).

TABLE III
THE NUMBER OF APPLICATIONS FOR VEHICLE DETECTION

METHOD	CALTECH	IR	PITTSBURGH	KITTI
ES	8134	7892	17872	19925
PVSP	707.7	375.67	618.54	880.22

quite useful in vehicle detection because "πHOG + PVSP + SSBT" outperforms "HOG + PVSP + SSBT." To avoid the performance degradation caused by the different vehicle sizes, multiple models can be considered. For example, two models are separately designed for sedans and SUVs and for buses and trucks, at the cost of increased computation time. Overall, the pair of πHOG and the PVSP demonstrates the best detection performance among all the possible combinations.

Finally, in order to determine the reduction in computation due to the use of PVSP, we compare ES and PVSP in terms of number of classifier applications. Obviously, the smaller is the number of applications used, the greater is the reduction. The result is summarized in Table III. As seen in Table III, ES requires many more applications of the classifiers than does PVSP, thereby wasting computational resources. In particular, the number of classifier applications in the Pittsburgh dataset is larger than those in the Caltech and IR datasets because the Pittsburgh dataset includes vehicles with various types and sizes. On average, PVSP achieves a 94.6% reduction in number of classifier applications from ES.

In Tables IV and V, various combinations are compared in terms of computation time. Different SSRs and HGs are used in Table IV, and different features are compared in Table V. In Table IV, the second column denotes the algorithms using HOG, while the fourth column denotes the algorithms using πHOG. The figure shows that PVSP is faster than other HGs in most datasets except SHDW and OP in the Caltech dataset. Overall, PVSP is the best among the SSRs and HGs. In Table V, different features are compared using ES and PVSP as search spaces.

In Table V, the second column denotes the algorithms using ES, while the fourth column denotes the algorithms using

TABLE IV
COMPUTATIONAL TIMES USING DIFFERENT SSRS/HGS (HOG/πHOG AS FEATURES, SVM AS A CLASSIFIER)

DATASET	METHOD	TIME (ms)	METHOD	TIME (ms)
Caltech	HOG+ES	251.18	πHOG+ES	613.04
	HOG+ SHDW	24.42	πHOG+ SHDW	35.85
	HOG+VH	122.99	πHOG+VH	239.69
	HOG+OP	113.44	πHOG+OP	156.74
	HOG+PVSP	78.7	πHOG+PVSP	166.77
IR	HOG+ES	95.53	πHOG+ES	521.95
	HOG+ SHDW	88.73	πHOG+ SHDW	132.96
	HOG+VH	79.75	πHOG+VH	146.53
	HOG+OP	116.7	πHOG+OP	182.16
	HOG+PVSP	49.97	πHOG+PVSP	89.04
Pittsburgh	HOG+ES	325.32	πHOG+ES	1332.0 8
	HOG+ SHDW	292.83	πHOG+ SHDW	448.8
	HOG+VH	122.64	πHOG+VH	224.37
	HOG+OP	294.35	πHOG+OP	486.5
	HOG+PVSP	109.64	πHOG+PVSP	204.1
Kitti	HOG+ES	360.5	πHOG+ES	1392.9
	HOG+ SHDW	208.48	πHOG+ SHDW	328
	HOG+VH	280.87	πHOG+VH	517.8
	HOG+PVSP	113.38	πHOG+PVSP	248.74

PVSP. As expected, πHOG takes longer than HOG but less time than LBP or LGP. PVSP reduces the computation time. Thus, the proposed method (πHOG + PVS) is the best choice because it demonstrates the best performance among the competing algorithms while requiring similar or less time than the other methods.

Concerning the classifier, SVM takes less time than ELM and kNN. Thus, considering not only detection performance, but also computation time, the linear SVM is a good choice. In conclusion, the combination of "πHOG + PVSP + SVM" is the best combination for vehicle detection.

VI. CONCLUSION

In this paper, a feature named πHOG and a new SSR method named PVSP were proposed to improve vehicle detection performance and reduce computational time, respectively, for complicated urban road environments. The proposed features and SSR methods demonstrated better performance using four datasets than the existing methods. The advantage of the PVSP is simultaneous reducing in computational time and reduced false positive rate. This paper focused not on a specific classifier, but on a new feature and SSR. Thus, the proposed method can be combined with other strong classifiers, such as the deformable part model [18]. The combination with various classifiers is recommended as a future work.

TABLE V
COMPUTATIONAL TIMES USING DIFFERENT FEATURES (ES/PVSP AS SSRS AND SVM AS A CLASSIFIER)

DATASET	METHOD	TIME (ms)	METHOD	TIME (ms)
Caltech	HOG+ES	251.18	HOG+PVSP	78.7
	ACF+ES	190.2	ACF+PVSP	145.56
	LBP+ES	736.82	LBP+PVSP	550.09
	LGP+ES	834.55	LGP+PVSP	685.55
	πHOG+ES	613.04	πHOG+PVSP	166.77
	πHOG+ES +ELM	3034.43	πHOG+PVSP +ELM	404.27
	πHOG+ES +kNN	8min	πHOG+PVSP +kNN	45476
IR	HOG+ES	95.53	HOG+PVSP	49.97
	ACF+ES	163.13	ACF+PVSP	100.26
	LBP+ES	565.34	LBP+PVSP	376.46
	LGP+ES	673.81	LGP+PVSP	507.4
	πHOG+ES	521.95	πHOG+PVSP	89.04
	πHOG+ES +ELM	1920.74	πHOG+PVSP +ELM	206.63
	πHOG+ES +kNN	3min	πHOG+PVSP +kNN	15297
Pittsburgh	HOG+ES	325.32	HOG+PVSP	109.64
	ACF+ES	531.39	ACF+PVSP	301.83
	LBP+ES	1961.9	LBP+PVSP	942.6
	LGP+ES	2327.74	LGP+PVSP	1128.5
	πHOG+ES	1332.08	πHOG+PVSP	204.1
	πHOG+ES +ELM	14513.77	πHOG+PVSP +ELM	761
	πHOG+ES +kNN	40min	πHOG+PVSP +kNN	103978
Kitti	HOG+ES	360.5	HOG+PVSP	113.38
	ACF+ES	454.28	ACF+PVSP	207.57
	LBP+ES	3792	LBP+PVSP	888.39
	LGP+ES	2901.95	LGP+PVSP	883.47
	πHOG+ES	1392.96	πHOG+PVSP	248.74
	πHOG+ES +ELM	36587	πHOG+PVSP +ELM	876.25
	πHOG+ES +kNN	779475	πHOG+PVSP +kNN	158334

APPENDIX

Assume that a prior is given by

$$\begin{aligned} p(\boldsymbol{\beta}, \phi|\mathbf{W}^{n-1}) &= \mathcal{NG}(\boldsymbol{\mu}^{n-1}, \mathbf{S}^{n-1}, \alpha^{n-1}, \lambda^{n-1}) \\ &\propto (\phi)^{\alpha^{n-1}} \exp\left[-\frac{\phi}{2}\left\{2\lambda^{n-1} + (\boldsymbol{\beta}-\boldsymbol{\mu}^{n-1})^T(\mathbf{S}^{n-1})^{-1} \times (\boldsymbol{\beta}-\boldsymbol{\mu}^{n-1})\right\}\right]. \end{aligned}$$

If the n-th vehicle $\mathbf{w}^n = (w_x^n \quad w_y^n \quad w_s^n)^T$, which respects the PVSP, is detected, the likelihood of $\boldsymbol{\beta}$ and ϕ are represented by

$$\begin{aligned} p(\mathbf{w}^n|\boldsymbol{\beta}, \phi, \mathbf{W}^{n-1}) &= \mathcal{N}\left(w_s^n|(\mathbf{w}_y^n)^T\boldsymbol{\beta}, \frac{1}{\phi}\right) \\ &\propto \phi^{\frac{1}{2}} \exp\left\{-\frac{\phi}{2}\left(w_s^n - (\mathbf{w}_y^n)^T\boldsymbol{\beta}\right)^2\right\}. \end{aligned}$$

Using the Bayes rule, the posterior after detecting the n-th vehicle $\mathbf{w}^n$ is computed as

$$\begin{aligned} p(\boldsymbol{\beta}, \phi|\mathbf{W}^n) &\propto \text{likelihood} \times \text{prior} \\ &= p(\mathbf{w}^n|\boldsymbol{\beta}, \phi, \mathbf{W}^{n-1}) p(\boldsymbol{\beta}, \phi|\mathbf{W}^{n-1}) \\ &\propto (\phi)^{\alpha^{n-1}+\frac{1}{2}} \exp\left[-\frac{\phi}{2}\left\{2\lambda^{n-1} + (\boldsymbol{\beta}-\boldsymbol{\mu}^{n-1})^T (\mathbf{S}^{n-1})^{-1}\right.\right. \\ &\qquad \left.\left. \times(\boldsymbol{\beta}-\boldsymbol{\mu}^{n-1}) + \left(w_s^n - \left(\mathbf{w}_y^n\right)^T \boldsymbol{\beta}\right)^2\right\}\right]. \end{aligned} \tag{A.1}$$

The part inside the bracket in (A.1)

$$\begin{aligned} f(\boldsymbol{\beta}) = 2\lambda^{n-1} &+ (\boldsymbol{\beta}-\boldsymbol{\mu}^{n-1})^T (\mathbf{S}^{n-1})^{-1} (\boldsymbol{\beta}-\boldsymbol{\mu}^{n-1}) \\ &+ \left(w_s^n - \left(\mathbf{w}_y^n\right)^T \boldsymbol{\beta}\right)^2 \end{aligned} \tag{A.2}$$

is quadratic in $\boldsymbol{\beta}$. By calculating a square term and performing some manipulation, we can rewrite (A.1) as

$$f(\boldsymbol{\beta}) = 2\lambda^n + (\boldsymbol{\beta}-\boldsymbol{\mu}^n)^T (\mathbf{S}^n)^{-1} (\boldsymbol{\beta}-\boldsymbol{\mu}^n) \tag{A.3}$$

where

$$\begin{aligned} (\mathbf{S}^n)^{-1} &= \left\{(\mathbf{S}^{n-1})^{-1} + \mathbf{w}_y^n \left(\mathbf{w}_y^n\right)^T\right\} \\ \boldsymbol{\mu}^n &= \mathbf{S}^n \left\{(\mathbf{S}^{n-1})^{-1} \boldsymbol{\mu}^{n-1} + \mathbf{w}_y^n w_s^n\right\}. \end{aligned}$$

Substituting $\boldsymbol{\beta} = \mathbf{0}$ into (A.2) and (A.3) and equating them yields

$$\begin{aligned} \lambda^n = \lambda^{n-1} + \frac{1}{2}\left\{(\boldsymbol{\mu}^{n-1})^T (\mathbf{S}^{n-1})^{-1} \boldsymbol{\mu}^{n-1} + (w_s^n)^2 \right. \\ \left. - (\boldsymbol{\mu}^n)^T (\mathbf{S}^n)^{-1} (\boldsymbol{\mu}^n)\right\}. \end{aligned}$$

Then, the posterior $p(\boldsymbol{\beta}, \phi|\mathbf{W}^{n-1})$ becomes another normal-gamma distribution

$$\begin{aligned} p(\boldsymbol{\beta}, \phi|\mathbf{W}^n) &= \mathcal{NG}(\boldsymbol{\mu}^n, \mathbf{S}^n, \alpha^n, \lambda^n) \\ &\propto (\phi)^{\alpha^n} \exp\left[-\frac{\phi}{2}\left\{2\lambda^n + (\boldsymbol{\beta}-\boldsymbol{\mu}^n)^T (\mathbf{S}^n)^{-1} (\boldsymbol{\beta}-\boldsymbol{\mu}^n)\right\}\right] \end{aligned}$$

where

$$\alpha^n = \alpha^{n-1} + \frac{1}{2}.$$

REFERENCES

[1] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, May 2006.

[2] Z. Sun, G. Bebis, and R. Miller, "Monocular precrash vehicle detection: Features and classifiers," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 2019–2034, Jul. 2006.

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

[4] G. Y. Song, K. Y. Lee, and J. W. Lee, "Vehicle detection by edge-based candidate generation and appearance-based classification," in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 428–433.

[5] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.

[6] A. Broggi, P. Cerri, and P. C. Antonello, "Multi-resolution vehicle detection using artificial vision," in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 310–314.

[7] A. Bensrhair *et al.*, "A cooperative approach to vision-based vehicle detection," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2001, pp. 207–212.

[8] D. Guo, T. Fraichard, M. Xie, and C. Laugier, "Color modeling by spherical influence field in sensing driving environment," in *Proc. IEEE Intell. Veh. Symp.*, 2000, pp. 249–254.

[9] C. Tzomakas and W. Seelen, Vehicle detection in traffic scenes using shadows," Inst. Neuroinformatik, Ruhr-Univ., Bochum, Germany, Tech. Rep. 98-06, 1998.

[10] Y. Feng and C. Xing, "A new approach to vehicle positioning based on region of interest," in *Proc. IEEE ICSESS*, 2013, pp. 471–474.

[11] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time precrash vehicle detection system," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 2002, pp. 171–176.

[12] R. O'Malley, E. Jones, and M. Glavin, "Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 453–462, Jun. 2010.

[13] C. H. Lee, Y. C. Lim, S. Kwon, and J. H. Lee, "Stereo vision-based vehicle detection using a road feature and disparity histogram," *Opt. Eng.*, vol. 50, no. 2, Feb. 2011, Art. ID. 027004.

[14] M. Bertozzi and A. Broggi, "Vision-based vehicle guidance," *Computer*, vol. 30, no. 7, pp. 49–55, Jul. 1997.

[15] A. Giachetti, M. Campani, and V. Torre, "The use of optical flow for road navigation," *IEEE Trans. Robot. Autom.*, vol. 14, no. 1, pp. 34–48, Feb. 1998.

[16] X. Ji, Z. Wei, and Y. Feng, "Effective vehicle detection technique for traffic surveillance systems," *J. Vis. Commun. Image Represent.*, vol. 17, no. 3, pp. 647–658, Jun. 2006.

[17] W. C. Chang and C. W. Cho, "Online boosting for vehicle detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 892–902, Jun. 2010.

[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[19] F. Porikli, "Integral histogram: a fast way to extract histograms in Cartesian spaces," in *Proc. IEEE Conf. Compu. Vis. Pattern Recog.*, 2005, vol. 1, pp. 829–836.

[20] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[21] A. Haselhoff and A. Kummert, "A vehicle detection system based on Haar and Triangle features," in *Proc. IEEE Intell. Veh. Symp.*, 2009, pp. 261–266.

[22] P. Sudowe and B. Leibe, "Efficient use of geometric constraints for sliding-window object detection in video," in *Proc. 8th Int. Conf. Comput. Vis. Syst.*, 2011, pp. 11–22.

[23] D. Gerónimo, A. Sappa, A. López, and D. Ponsa, "Adaptive image sampling and windows classification for on-board pedestrian detection," in *Proc. 5th Int. Conf. Comput. Vis. Syst.*, 2007, pp. 1–10.

[24] J. Kim, J. Baek, and E. Kim, "On-road vehicle detection based on effective hypothesis generation," in *Proc. 22nd IEEE Int. Symp. RO-MAN Interactive Commun.*, 2013, pp. 252–257.

[25] P. Geismann and G. Schneider, "A two-staged approach to vision-based pedestrian recognition using Haar and HOG Features," in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 554–559.

[26] M. Pedersoli, J. Gonzalez, and A. D. Bagdanov, "Efficient discriminative multiresolution cascade for real-time human detection applications," *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1581–1587, Oct. 2011.

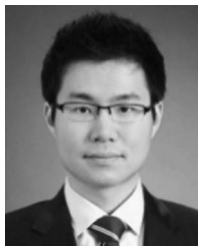
[27] G. Gualdi, A. Prati, and R. Cucchiara, "Multi-stage particle windows for fast and accurate object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1589–1604, Aug. 2012.

[28] Q. Yuan and V. Ablavsky, "Learning a family of detectors via multiplicative kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 514–530, Mar. 2011.

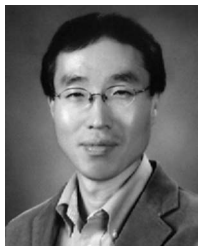
[29] Z. M. Qian, H. X. Shi, and J. K. Yang, "Video vehicle detection based on local feature," *Adv. Mater. Res.*, vol. 186, pp. 55–60, 2011.

[30] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, "A cascade of boosted generative and discriminative classifiers for vehicle detection," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, Jan. 2008, Art. ID. 782 432.
[31] B. Jun and D. Kim, "Robust face detection using local gradient patterns and evidence accumulation," *Pattern Recognit.*, vol. 45, no. 9, pp. 3304–3316, Sep. 2012.
[32] J. Zhang, K. Huang, Y. Yu, and T. Tan, "Boosted local structured HOG-LBP for object localization," in *Proc. IEEE Conf. CVPR*, 2011, pp. 1393–1400.
[33] G. Ratsch, T. Onoda, and K. R. Muller, "Soft margins for AdaBoost," *Mach. Learn.*, vol. 42, no. 3, pp. 287–320, Mar. 2001.
[34] V. Vapnik, *The Nature of Statistical Learning Theory.* New York, NY, USA: Springer-Verlag, 1995.
[35] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machines: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006.
[36] P. Cunningham and S. J. Delany, "k-Nearest neighbor classifiers," in *Multiple Classifier Systems*. Berlin, Germany: Springer-Verlag, 2007.
[37] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, vol. 1. Boston, MA, USA: PWS-Kent, 1996.
[38] F. Han, Y. Shan, R. Cekander, H. S. Sawhney, and R. Kumar, "A two stage approach to people and vehicle detection with HOG-based SVM," in *Proc. Perform. Metrics Intell. Syst. Workshop*, 2006, pp. 133–140.
[39] L. Bourdev and J. Brandt. "Robust object detection via soft cascade," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, vol. 2, pp. 236–243.
[40] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[41] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, Aug. 2006.
[42] C. Walck, *Handbook on Statistical Distributions for Experimentalists*. Stockholm, Sweden: Univ. of Stockholm Press, 2000.
[43] O. L. Junior and U. Nunes, "Improving the generalization properties of neural networks: An application to vehicle detection," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, 2008, pp. 310–315. [Online]. Available: http://www.vision.caltech.edu/archive.html
[44] National Motor Vehicle Crash Causation Survey, National Highway Traffic Safety Administration, Washington, DC, USA, 2008. [Online]. Available: http://users.ece.cmu.edu/~hyunggic/vehicleDPM.html
[45] KITTI, Object Detection and Orientation Estimation Benchmark. [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_object.php
[46] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[47] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[48] J. Hwang, K. Rou, S. Park, E. Kim, and H. Kang, "PCA based vehicle detection system for ACC," in *Proc. ICEIC*, 2006, pp. 182–187.
[49] J. Baek, S. Hong, J. Kim, and E. Kim, "Bayesian learning of a search region for pedestrian detection," *Multimedia Tools Appl.*, to be published, DOI: 10.1007/s11042-014-2329-z.

Jisu Kim received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2011, where he is currently working toward a combined master's and doctoral degree. He has studied machine learning and pattern recognition for vehicle detection.



Jeonghyun Baek received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2011, where he is currently working toward a combined master's and doctoral degree. He has studied machine learning, computer vision, and optimization for pedestrian detection.



Euntai Kim was born in Seoul, Korea, in 1970. He received the B.S., M.S., and Ph.D. degrees from Yonsei University, Seoul, Korea, in 1992, 1994, and 1999, respectively, all in electronic engineering. From 1999 to 2002, he was a Full-Time Lecturer with the Department of Control and Instrumentation Engineering, Hankyong National University, Anseong, Korea. He was a Visiting Scholar with the University of Alberta, Edmonton, AB, Canada, in 2003 and was also a Visiting Researcher with the Berkeley Initiative in Soft Computing, University of California, Berkeley, CA, USA, in 2008. Since 2002, he has been with the faculty of the School of Electrical and Electronic Engineering, Yonsei University, where he is currently a Professor. His current research interests include computational intelligence and statistical machine learning and their application to intelligent robotics, unmanned vehicles, and robot vision.