Vehicle Detection in High-Resolution Aerial Images via Sparse Representation and Superpixels

Ziyi Chen, Cheng Wang, *Member, IEEE*, Chenglu Wen, *Member, IEEE*, Xiuhua Teng, Yiping Chen, Haiyan Guan, Huan Luo, Liujuan Cao, and Jonathan Li, *Senior Member, IEEE*

***Abstract*—This paper presents a study of vehicle detection from high-resolution aerial images. In this paper, a superpixel segmentation method designed for aerial images is proposed to control the segmentation with a low breakage rate. To make the training and detection more efficient, we extract meaningful patches based on the centers of the segmented superpixels. After the segmentation, through a training sample selection iteration strategy that is based on the sparse representation, we obtain a complete and small training subset from the original entire training set. With the selected training subset, we obtain a dictionary with high discrimination ability for vehicle detection. During training and detection, the grids of histogram of oriented gradient descriptor are used for feature extraction. To further improve the training and detection efficiency, a method is proposed for the defined main direction estimation of each patch. By rotating each patch to its main direction, we give the patches consistent directions. Comprehensive analyses and comparisons on two data sets illustrate the satisfactory performance of the proposed algorithm.**

***Index Terms*—Aerial image, high resolution, sparse representation, superpixel, vehicle detection.**

I. Introduction

DUE to economic development and an increasing demand for fast and convenient travel, automobiles have become extremely popular and play an important role in daily life. The large number of cars generates heavy pressure on transportation, road, and traffic regulatory authorities and also makes vehicle monitoring a vital part of traffic information gathering, traffic jam and congestion prevention, traffic accident control, vehicle flow statistics, road network planning, and estimating parking situations [1]–[5].

A large number of fixed ground sensors, such as induction loops, bridge sensors, stationary cameras, and radar sensors, are required to efficiently monitor vehicles and gather traffic information [6], [7]. By using these fixed ground sensors, the traffic flow, vehicle density, and parking situation are partially acquired. However, these methods fail to provide a complete overview of the traffic situation, which is a vital information source for studying road network planning, modeling, optimization, and traffic-related statistics.

The demand for gathering an overview of traffic situations leads to monitoring of vehicles via alternate methods such as remote sensing images captured by satellites or airplanes. Due to their capability to provide full coverage of an area of interest, remote sensing images have been widely applied for monitoring vehicles [6], [8], [9]. Currently, there are many commercial Earth observation satellites such as IKONOS, GeoEye, WorldView-2, WorldView-3, and QuickBird that provide publicly available images with the spatial resolution of a submeter. Benefiting from the high spatial resolution, satellite images are a data source for studying vehicle monitoring [5], [7], [10]. Compared with satellite images, aerial images are usually preferred because of their higher spatial resolution ranging from 0.1 to 0.5 m [11], [12] and their easier data acquisition [13]. With high spatial resolution, vehicles, even small cars, can be clearly identified in aerial images. Thus, detecting vehicles from high-resolution aerial images is attractive for traffic monitoring and mitigation over a large area [14]. Manually detecting vehicles from aerial images is time consuming and labor intensive. Therefore, it is imperative to develop an automatic vehicle detection method from high-spatial-resolution aerial images.

Conversely, automatically detecting vehicles from high-resolution aerial images is still a challenging task because the presence of a large number of structures (e.g., trash bins, road marks, electrical units, and air conditioning units on top of buildings), particularly in urban areas, can cause false alarms. In addition, the partial occlusions caused by the shadows of trees and buildings might greatly increase the difficulties of vehicle detection. The illumination condition is another critical factor for detecting vehicles from aerial images. Training samples play an important role in object recognition. In order to obtain high classification accuracy between vehicles and background, a training sample set that contains kinds of positives and negatives is required. The simplest way is to use

Manuscript received January 17, 2015; revised April 17, 2015 and June 3, 2015; accepted June 26, 2015. This work was supported in part by the National Natural Science Foundation of China under Project 61371144. *(Corresponding author: Cheng Wang.)*

Z. Chen, C. Wang, C. Wen, H. Luo, and L. Cao are with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen 361005, China (e-mail: cwang@xmu.edu.cn).

X. Teng is with the School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350014, China.

Y. Chen is with the School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China, and also with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen 361005, China.

H. Guan is with the College of Geography and Remote Sensing, Nanjing University of Information Science and Technology, Nanjing 210044, China.

J. Li is with the Key Laboratory of Underwater Acoustic Communication and Marine Information Technology of the Ministry of Education, Xiamen University, Xiamen 361005, China, and also with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

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/TGRS.2015.2451002

the whole training sample set. However, the whole training sample set is usually too large and redundant, which causes high computational complexity of training or detection. Thus, to train a classifier, it is necessary to select a small and complete subset of training samples. However, it is time consuming and difficult to manually select all of the representative samples from a large number of negatives. Additionally, both the manual and random training sample selection methods cannot promise to train an optimal classifier to obtain the best performance.

To improve the detection efficiency and automatically construct a complete and representative training set, we develop an algorithm using sparse representation and superpixel segmentation for automatic vehicle detection in high-resolution aerial images. To effectively slide the detection window without side effects, a superpixel-based segmentation is introduced to segment the high-resolution aerial image into a set of superpixels. Based on the centers of superpixels, meaningful patches are generated accordingly. Then, sparse representation is applied for dictionary learning and classification processing. To construct an optimal training subset, we propose an iteration of sample selection strategy based on sparse representation. During the training sample selection, the completeness of representative positives and negatives are both considered. With the selected optimal training set, we obtain a sparse representation dictionary with high discriminative ability for vehicle detection.

We apply our method to two high-resolution aerial image data sets. One data set is the aerial images covering the city of Toronto, Canada, with 0.15-m spatial resolution; the other data set is from the overhead imagery research data set (OIRDS). Experimental analyses and comparisons on both data sets demonstrate the superior performance of our method versus several state-of-the-art methods, including histogram of oriented gradient (HOG) + linear support vector machine (SVM) [15], [16], scale-invariant feature transform (SIFT) + linear SVM [17], and HOG + kernel SVM [18].

II. Related Work

Sparse representation and superpixel segmentation have received considerable attention in computer vision [19]–[22]. Sparse representation has been successfully applied in many fields, including face recognition, object classification, image classification, image de-noising, image restoration, visual saliency, and data compression [23]–[30]. Yokoya and Iwasaki applied sparse representation for object detection in remote sensing images and achieved good results [31]. The development of superpixel segmentation provides a new way for image preprocessing, image segmentation, feature extraction, and object tracking [22], [32]. In recent years, much research has focused on superpixel-based image segmentation, and many approaches have been developed. Representative approaches include graph-based algorithms and gradient-based algorithms. The latest achievements are simple linear iterative clustering (SLIC) [33], edge-weighted centroidal Voronoi tessellations-based (VCells) [34], and entropy-rate clustering [35]. Using sparse representation and superpixel segmentation is a new way to detect vehicles from high-resolution aerial images.

Many approaches have been developed for vehicle detection from high-resolution aerial images [6], [8], [11]–[14], [36]–[44]. Most of the approaches can be separated into two types of vehicle models, i.e., appearance-based implicit models and explicit models.

Appearance-Based Implicit Models: An appearance-based implicit model typically consists of image intensity or texture features computed using a small window or kernel that surrounds a given pixel or a small cluster of pixels. Then, detection is conducted by examining feature vectors of the image's immediate surrounding pixels [14]. Cheng *et al.*, using dynamic Bayesian networks for vehicle detection from aerial surveillance, achieved promising results on a challenging data set [43]. However, the color model, specially designed for separating cars from the background, still cannot avoid false and missing detection due to the overlap of the cars' and the background's color models. Another problem is that the approach must remerge the detected pixels into individual vehicles, which is a difficult task when vehicles are parked in close proximity. Additionally, detection checking over all of the pixels increases not only the computational complexity but also the false detection rate. Shao *et al.* first explored vehicle detection by using multiple features (e.g., HOG, local binary pattern, and opponent histogram) and the intersection kernel SVM [45]. Similarly, Moranduzzo and Melgani combined the SIFT and SVM for detecting cars from unmanned aerial vehicle (UAV) images [17]. Kembhavi *et al.* detected cars from high-resolution aerial images by applying partial least squares, a powerful feature selection analysis, and a redundant feature descriptor, consisting of color probability maps, HOG features, and pairs of pixel comparisons that catch a car's structural features [11]. Their work shows an impressive performance. Moranduzzo and Melgani proposed a catalog-based approach for detecting cars in UAV images [44]. However, its application is limited to special scenes because it must use the asphalt area as an *a priori* guide. Another problem is that it must also remerge the detected pixels into individual vehicles. Hinz and Baumgartner extracted vehicle features based on a hierarchical vehicle model, which details different levels [36]. Khan *et al.* extracted vehicle features based on a 3-D model [41]. Wang *et al.* applied the implicit shape model and Hough voting for car detection in 3-D point clouds, with impressive results [46].

Explicit Models: Regarding the explicit model, a vehicle is usually described by a box, a wireframe representation, or a morphological model. Car detection is performed by matching a car model to the image with a "top-down" or a "bottom-up" strategy [14]. Zheng *et al.* utilized grayscale opening transformation and grayscale top-hat transformation to identify potential vehicles in the light or white background and used grayscale closing transformation and grayscale bot-hat transformation to identify potential vehicles in the black or dark background. Then, size information is employed to eliminate false alarms [14]. Their approach exhibits good performance on highway aerial images; however, the gray value estimates of the background and geographic information system data are required. As a result, this method is not suitable for general scenes. A vehicle has been also represented as a 3-D box with dimensions for width, length, and height in [47].

Several studies have also studied the sample selection from a large amount of training data. Zhou *et al.* [48] proposed a sample reduction method to address the sample unbalance problem of SVM. Nie *et al.* [49] proposed an active method to select the most representative samples for labeling in the early active learning stage to reduce manual works. They used an iteration method to select a subset of the most representative samples by using structured sparsity-inducing norms. However, the method is still time consuming.

Generally speaking, two disadvantages of car detection methods using an explicit model are obvious. First, the detection is usually based on detected edges, leading to unrobustness to noise and a complex background. Second, these methods have a poor performance under the situations of slight occlusions and shape variations because car models are rigidly predefined. Most state-of-the-art car detection methods (even commercial software) for remote sensing images use an implicit model because of its better generalization ability [7].

However, existing methods employing the implicit model still suffer from the following two problems.

First, most methods are pixel based or use a slide window with a predefined slide step during detection. The pixel-based methods are computationally intensive. In addition, these methods must remerge the detected pixels into individual vehicles, which is a difficult task when vehicles are parked in close proximity. In slide window methods, the slide step influences the detection recall rate and the processing speed. A large slide step may result in fast processing speed but cause a decrease in the recall rate. A small slide step may increase the recall rate but lead to a high computation cost. It is difficult to trade off the detection recall rate and the processing speed. A more effective scanning strategy is desired for improving the scanning efficiency.

Second, the training samples are manually or even randomly selected. Manual training sample selection is time consuming. For vehicle detection in aerial images, a complex background results in a large number of negatives, making it difficult to manually select an optimal negative training subset. With regard to the random selection method, it might cause an unstable detection performance and usually cannot achieve an optimal performance. An effective training sample selection method needs to be developed.

Consequently, a strong need exists to exploit a solution for the two problems above.

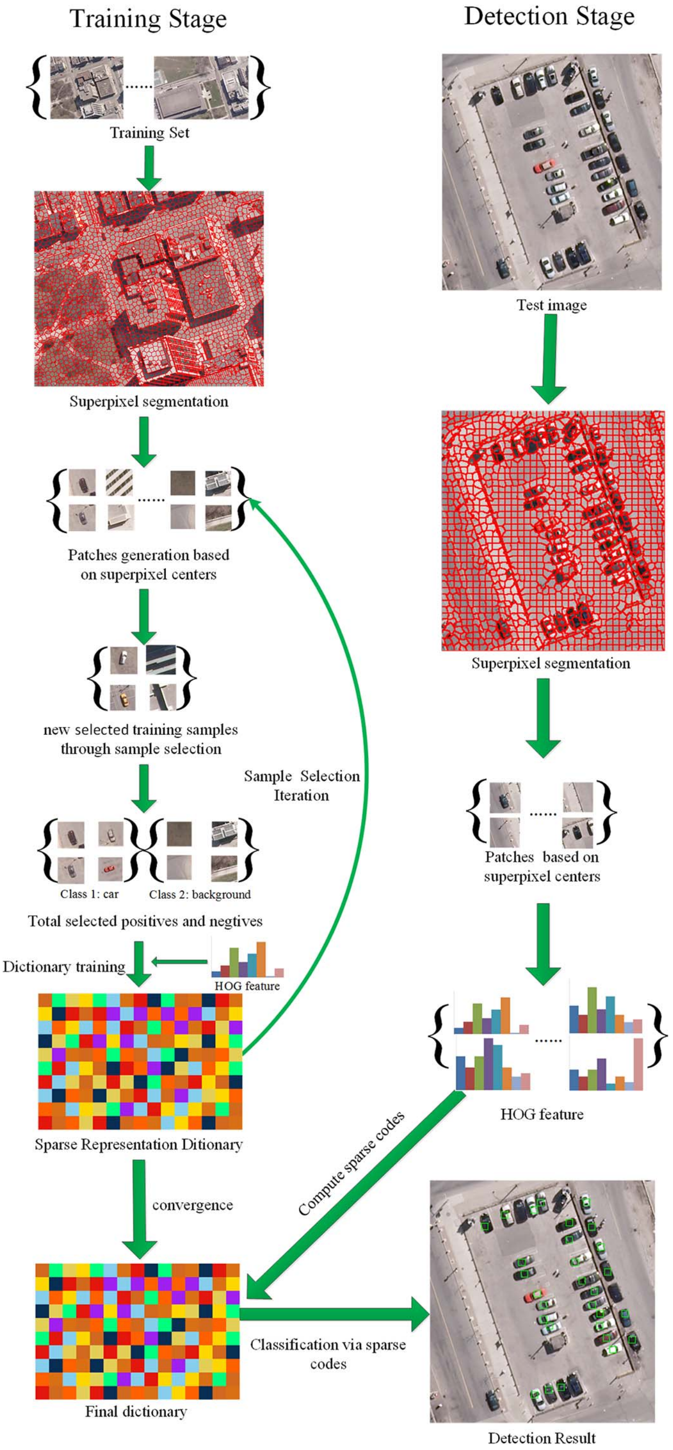

Training Stage
Detection Stage
Training Set
Superpixel segmentation
Patches generation based on superpixel centers
new selected training samples through sample selection
Sample Selection Iteration
Class 1: car
Class 2: background
Total selected positives and negtives
Dictionary training
HOG feature
Sparse Representation Ditionary
convergence
Final dictionary
Test image
Superpixel segmentation
Patches based on superpixel centers
HOG feature
Compute sparse codes
Classification via sparse codes
Detection Result

Fig. 1. Framework of the proposed method includes two stages: training stage and detection stage.

III. Proposed Solution

A. Framework

As shown in Fig. 1, the framework of our proposed method includes two stages, i.e., dictionary training and car detection. In the training stage, training images are first segmented into superpixels via the proposed superpixel generation method. Based on the superpixel centers, we generate the meaningful patches as the whole training set. Then, we select a small training subset to initialize a small sparse dictionary. In our method, the grids of HOG descriptors of patches are extracted as the dictionary input. With the trained dictionary, we estimate the similarity between the remaining training samples and cars. The negatives with the highest similarity and the positives with the lowest similarity are selected to add into the training subset to train a new dictionary for the next sample selection iteration. The training sample selection iteration proceeds until convergence is reached. In this paper, two situations are regarded as convergence conditions. First, the trained dictionary contains more than 2000 items. Second, the classification accuracy of test images is higher than 80% under a 0.7 recall rate. Once converged, we apply the sparse representation dictionary to detect vehicles.

In the detection stage, a test image is first segmented into superpixels, based on the center of which we scan the test image with high efficiency. According to the sparse codes during scanning, the patch candidates are classified into cars and the background.

B. Superpixel Segmentation

In our proposed method, superpixel segmentation is an important step. Superpixel segmentation breakage, defined as the disconnection of segmentations, affects the scanning location accuracy and detection performance. To obtain superpixel segmentation with low breakage, we proposed a superpixel segmentation method designed specifically for our framework.

Given a uniform partition $P = \{p_i\}_{i=1}^{j}$ of image $C = \{r(e), g(e), b(e)\}_{e \in C}$, where j represents the initial partition number, $r(e)$, $g(e)$, and $b(e)$ represent the red (R), green (G), and blue (B) components of color space for pixel e, respectively. Then, each color center of partition p_i is calculated by using the following function:

$$\begin{aligned} R_{P_i} &= \frac{1}{|p_i|} \sum_{e \in p_i} R(e) \\ G_{P_i} &= \frac{1}{|p_i|} \sum_{e \in p_i} G(e) \\ B_{P_i} &= \frac{1}{|p_i|} \sum_{e \in p_i} B(e) \end{aligned} \tag{1}$$

where $|p_i|$ is the number of pixels in partition p_i. R_{P_i}, G_{P_i}, and B_{P_i} represent the color centers of the RGB components, respectively. In the next step, we iteratively update each partition's boundary pixels according to three measurements. The first measurement is the color distance between a boundary pixel e and a neighbor partition p_i's color center, which is defined as

$$d_c(e, p_i) = \frac{\sqrt{(r(e) - R_{p_i})^2 + (g(e) - G_{p_i})^2 + (b(e) - B_{p_i})^2}}{A_N} \tag{2}$$

where A_N is the normalization term to make the smallest color distance to be 1. Thus, A_N is set as the smallest color distance extracted from the distances between the boundary pixels and their corresponding neighbor partitions. Empirically, setting A_N at 25 is sufficient to obtain a good result.

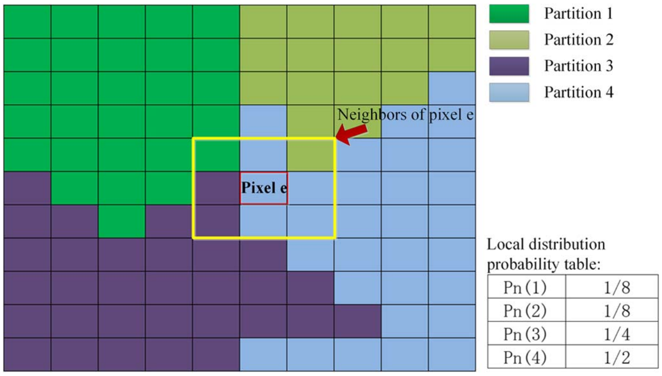


Fig. 2. Statistics distribution probability within pixel e's local 3 × 3 area.

The second measurement is the space distance between a boundary pixel e and its neighbor partitionp_i's space center. The space center of partition p_i is defined as

$$\begin{aligned} S_{P_i}(u, v) &= \left(\sum_{e \in P_i} w_e \cdot e_u, \sum_{e \in P_i} w_e \cdot e_v \right) \\ \hat{o}_e &= \frac{R_{P_i}}{|r(e) - R_{P_i}|} + \frac{G_{P_i}}{|g(e) - G_{P_i}|} + \frac{B_{P_i}}{|b(e) - G_{P_i}|} \\ o_e &= \frac{\hat{o}_e}{\sum_{e \in P_i} \hat{o}_e} \end{aligned} \tag{3}$$

where u and v are the position coordinates of pixel e in the image, $\hat{o}_e$ and o_e are the weights of pixel e, and o_e is the normalized term of $\hat{o}_e$. Then, the space distance from pixel e to its neighbor partition p_i is calculated as

$$d_s(e, p_i) = \frac{\sqrt{(u - u_{p_i})^2 + (v - v_{p_i})^2}}{B_N} \tag{4}$$

where B_N is a normalization term to make the smallest space distance to be 1. Thus, B_N is set as the smallest space distance extracted from the distances between the boundary pixels and their corresponding neighbor partitions. Empirically, setting B_N at 10 is sufficient to obtain a good result.

The third measurement is a boundary pixel e's local information that computes the statistical probability of pixels assigned to partition p_i within an area centered on e. The definition is as follows:

$$h(e, p_i) = \frac{N(e, p_i)}{L} \tag{5}$$

where $N(x, p_i)$ represents the number of pixels in partition p_i within e's local area, and L is the total pixel number within the defined e's local area. A large L improves the robustness of local information, but it increases the computational complexity. In our method, we use a 3 × 3 local area. As shown in Fig. 2, the yellow rectangle within a 3 × 3 window is defined as e's local area. We compute the statistical probabilities of pixels in the adjacent partitions of e (i.e., partitions 1, 2, 3, and 4).

Finally, the probability of $e \in p_i$ is calculated by

$$prob(e, p_i) = \frac{1}{d_c(e, p_i)} \cdot \frac{1}{d_s(e, p_i)} \cdot h(e, p_i). \tag{6}$$

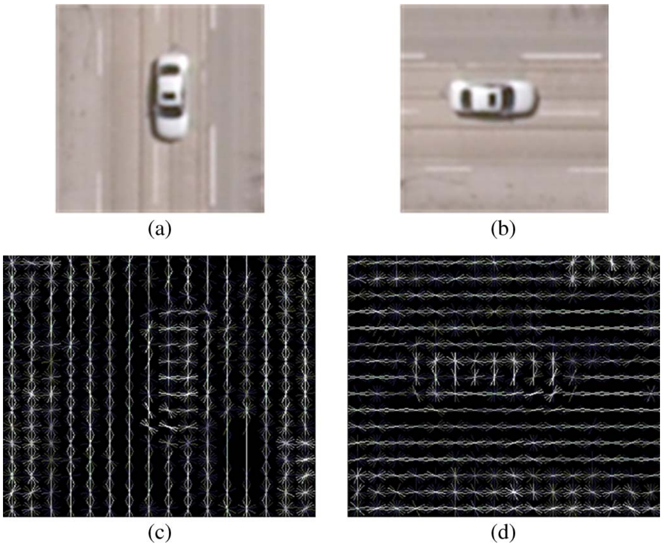


Fig. 3. (a) and (b) Two original image patches. (c) and (d) HOG descriptors of (a) and (b).

According to (6), we iteratively update boundary pixels of the segmented partitions until boundary pixels remained unchanged or reach the iteration termination condition. By using the local information, the segmented patches are smooth and well handled in disconnections. Another benefit from local information is that the robustness to noise is enhanced. The detailed analysis is discussed in the experimental part.

C. Grids of HOG Descriptor

In this paper, we use the grids of HOG descriptor to describe a patch. Here, we give a brief introduction about the grids of HOG descriptor proposed by Dalal and Triggs [16]. Given an image patch, we first divide the patch into small spatial regions ("cells"). For each cell, we accumulate a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. After this, the histogram entries of each cell are combined to form the representation of the patch. For better invariance to illumination, shadowing, etc., it is useful to contrast-normalize the local responses before using them. Fig. 3 shows an example of the grids of HOG descriptor. Fig. 3(a) and (b) are the same image patch with different orientations. We computed the HOG descriptors of the two patches with a cell size of 5 pixels. Fig. 3(c) and (d) are the HOG descriptors of Fig. 3(a) and (b), respectively. In Fig. 3, the descriptors of the two patches are different. As shown in Fig. 3, although it has achieved great success in object detection area, the grids of HOG is orientation sensitive. Notice that, instead of RGB images, we compute the HOG features based on grayscale images.

D. Sparse Representation

In our method, we apply the sparse representation method proposed by Jiang *et al.* [50] for sample selection, training, and testing.

Let $Y = [y_1, \ldots, y_N] \in R^{n \times N}$ denote N n-dimensional input signals. Then, learning a reconstructive dictionary with K items for sparse representation of Y can be accomplished by solving the following problem:

$$\langle D, X \rangle = \arg\min_{D,X} \|Y - DX\|_2^2 \quad \text{s.t.} \quad \forall\, i, \|x_i\|_0 \leq T \tag{7}$$

where T is a sparsity threshold, D is the sparse representation dictionary, and X represents the sparse codes. Equation (7) can be replaced by an l_1-norm problem

$$\langle D, X \rangle = \arg\min_{D,X} \|Y - DX\|_2^2 + \gamma \|X\|_1 \tag{8}$$

where γ is a parameter to balance the reconstruction error and the sparsity of representation codes. The equality of (7) and (8) was proved in [51]. The K singular value decomposition (K-SVD) algorithm [52] is an iterative approach to minimize the energy in (8) and learns a reconstructive dictionary for the sparse representation of signals. Reversely, given a dictionary D, the sparse representation x_i of an input signal y_i is computed as

$$x_i = \arg\min_{x} \|y_i - Dx\|_2^2 + \gamma \|x\|_1. \tag{9}$$

Due to the discrimination of the sparse codes among different classes, the sparse codes can be directly used for classification. The orthogonal matching pursuit algorithm [53] is used to solve (9). To increase the discriminability of the obtained sparse codes, a term representing the training samples' label information is added to the training dictionary. Thus, the objective function for dictionary construction is defined as

$$\begin{aligned}\langle D, A, X \rangle = \arg\min_{D,A,X} & \|Y - DX\|_2^2 \\ & + \alpha \|Q - AX\|_2^2 \quad \text{s.t.} \quad \forall\, i, \|x\|_0 < T \end{aligned} \tag{10}$$

where α controls the relative contribution between reconstruction and label consistency regularization, $Q = [q_1, \ldots, q_N] \in R^{K \times N}$ are the "discriminative" sparse codes of input signals Y for classification, and A is a linear transformation matrix that transforms the original sparse codes to be the most discriminative in sparse feature space R^K. The term $\|Q - AX\|_2^2$ represents the discriminative sparse code error, which forces signals from the same class to have very similar sparse representations and results in good classification performance. We say that $q_i = [q_i^1, \ldots, q_i^K]^t = [0, \ldots, 1, 1, \ldots, 0]^t \in R^K$ is a discriminative sparse code corresponding to an input signal y_i if the nonzero values of q_i occur at those indices where the input signal y_i and the dictionary item d_k share the same label. For example, assume that $D = [d_1, \ldots, d_6]$ and $Y = [y_1, \ldots, y_6]$, where y_1, y_2, d_1, and d_2 are from class 1; y_3, y_4, d_3, and d_4 are from class 2; and y_5, y_6, d_5, and d_6 are from class 3. Then, Q can be defined as

$$Q \equiv \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

where each column corresponds to a discriminative sparse code for an input signal.

During dictionary learning, a term that optimizes the discriminative power of sparse codes between different classes is included into the objective function of dictionary learning. Then, the objective function is represented as follows:

$$\langle D, W, A, X\rangle = \arg\min_{D,W,A,X} \|Y - DX\|_2^2 + \alpha\|Q - AX\|_2^2 + \beta\|H - WX\|_2^2 \quad \text{s.t.} \quad \forall\, i, \|x\|_0 < T \tag{11}$$

where the term $\|H - WX\|_2^2$ represents the classification error. W denotes the classifier parameters. $H = [h_1, \ldots, h_N] \in R^{m\times N}$ are the class labels of input signals Y. $h_i = [0, 0, \ldots, 1, \ldots, 0, 0]^t \in R^m$ is a label vector corresponding to an input signal y_i, where nonzero position indicates the class of y_i. m is the number of classes, and α and β are the scalars controlling the relative contribution of the corresponding terms.

To employ the K-SVD algorithm to solve (11), (11) is rewritten as follows:

$$\langle D, W, A, X\rangle = \arg\min_{D,W,A,X} \left\| \begin{pmatrix} Y \\ \sqrt{\alpha}Q \\ \sqrt{\beta}H \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\alpha}A \\ \sqrt{\beta}W \end{pmatrix} X \right\|_2^2 \quad \text{s.t.} \quad \forall\, i, \|x_i\|_0 \le T. \tag{12}$$

Denote $Y_{\text{new}} = (Y^t, \sqrt{\alpha}Q^t, \sqrt{\beta}H^t)^t$, and $D_{\text{new}} = (D^t, \sqrt{\alpha}A^t, \sqrt{\beta}W^t)^t$. Then, (12) is equal to the following function:

$$\langle D_{\text{new}}, X\rangle = \arg\min_{D_{\text{new}},X} \left\{\|Y_{\text{new}} - D_{\text{new}}X\|_2^2\right\} \quad \text{s.t.} \quad \forall\, i, \|x_i\|_0 \le T. \tag{13}$$

Equation (13) is just the form that the K-SVD algorithm solves. After we obtain $D=\{d_1, \ldots, d_K\}$, $A=\{a_1, \ldots, a_K\}$, and $W = \{w_1, \ldots, w_K\}$ from D_{new}, we cannot simply use D, A, and W for testing because D, A, and W are L_2-normalized jointly, i.e., $\forall\, k, \|d_k^t, \sqrt{\alpha}a_k^t, \sqrt{\beta}w_k^t\|_2 = 1$. Thus, the desired dictionary $\hat{D}$, the transform parameters $\hat{A}$, and the classifier parameters $\hat{W}$ are recomputed as follows:

$$\hat{D} = \left\{\frac{d_1}{\|d_1\|_2}, \ldots, \frac{d_K}{\|d_K\|_2}\right\}, \; \hat{A} = \left\{\frac{a_1}{\|d_1\|_2}, \ldots, \frac{a_K}{\|d_K\|_2}\right\}$$
$$\hat{W} = \left\{\frac{w_1}{\|d_1\|_2}, \ldots, \frac{w_K}{\|d_K\|_2}\right\}. \tag{14}$$

The desired $\hat{D}$, $\hat{A}$, and $\hat{W}$ are directly applied for tests. The final classification prediction l can be simply represented by

$$l = \hat{W}\hat{x}_i. \tag{15}$$

The label of y_i is regarded as the classification scores corresponding to each class. In our method, the grids of HOG features are utilized as the initial input signals Y. As shown in Fig. 4, we extract the grids of HOG feature of a car sample as the input test and solve the sparse codes of the test sample. According to the sparse codes, we compute the scores of the test patch corresponding to each class. Clearly, the test sample has the highest score for cars. Thus, we classify the test sample as car. The background classes of b1–b6 were constructed through our sample selection procedure and built according to the estimated similarities to a vehicle, instead of their actual background classes, such as grass, ground, and air conditioner.

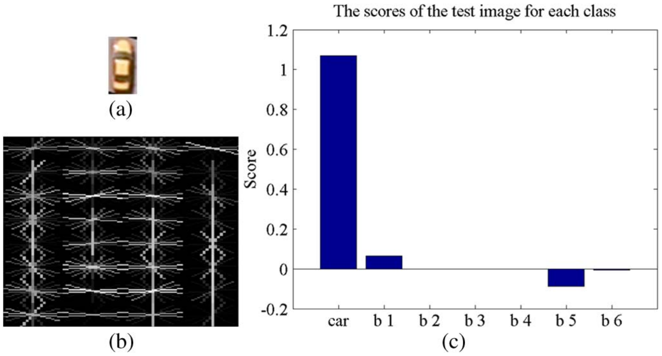


Fig. 4. (a) Test sample of car. (b) HOG feature of the test sample. (c) Class scores of the test sample responding to each class according to the sparse codes. b1 to b6 are background classes.

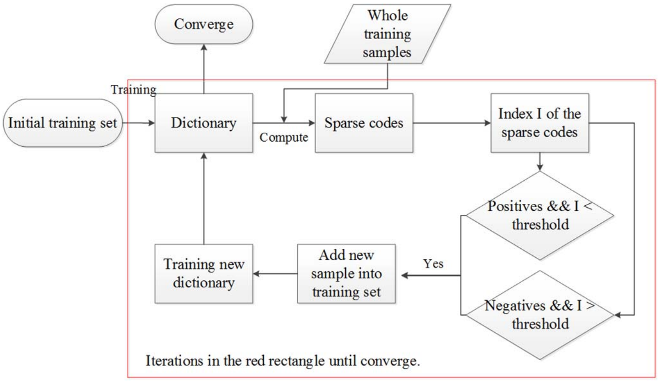

Converge
Whole training samples
Training
Initial training set
Dictionary
Compute
Sparse codes
Index I of the sparse codes
Positives && I < threshold
Training new dictionary
Add new sample into training set
Yes
Negatives && I > threshold
Iterations in the red rectangle until converge.

Fig. 5. Framework of the proposed training sample selection method.

E. Iterative Training Sample Selection

In this section, we introduce an automatic training sample selection approach that considers the difference of interclass and the completeness of intraclass to construct a compact training set to improve training and classification efficiency.

Fig. 5 shows the framework of the iterative training sample selection method. First, we manually select several dozen samples of cars and background samples to initialize the training set. Second, the grids of HOG features are extracted from the training samples as the input signal Y for dictionary training, according to (11). Third, the sparse codes of all the other training samples are calculated. Each sample is assigned a class distribution index. Based on this index, we select samples to add into the training set. The class distribution index I of a sample is defined as follows:

$$I = \frac{l_i}{\sum_{i=1}^{m} |l_i|} \tag{16}$$

where l is the label information of samples computed from (15), and m is the number of total classes. According to the class distribution index, we estimate the similarity between a sample and a car. Fourth, the samples are selected to join the training set according to the following two criteria.

(1) A positive sample, whose value I is lower than the defined threshold, will be selected as a new positive sample. The threshold starts from a small threshold (0.1 in our method). Then, it continually increases during the sample selection iteration.

(2) A negative sample, whose value I is higher than the defined threshold, will be selected as a new negative sample. The threshold starts from a large threshold (0.9 in our method). Then, it continually decreases during the sample selection iteration.

The first criterion ensures the difference of the intraclass car samples, and the second criterion ensures the discriminability of the selected training samples for classifying the cars and the background. To forbid the reselection of samples, we label all the unselected samples as 0 and the selected samples as 1. Fifth, a new and larger dictionary is trained according to the new training subset. We iteratively run these steps to select samples until the aforementioned convergence conditions are reached.

As aforementioned, the grids of HOG descriptor used in our method are orientation sensitive. The orientation of the vehicle is unknown in the test image. Thus, the test image should be scanned at multiple rotations [11]. Usually, each test patch is rotated with an angle interval of 5° (or 30°, 45°, etc.) for scanning, resulting in a dramatic decrease in detection efficiency. To further improve detection efficiency, we estimate the main direction of each patch and automatically rotate all patches to their main directions to maintain orientation consistence during dictionary training and vehicle detection. Through main direction estimation, we only need to examine each test patch in one orientation. Thus, we improve the detection efficiency. Considering that a car patch usually contains two long straight lines along the car's length direction, we define the main direction of a patch as the direction of straight lines with a longer length than the predefined threshold. In this paper, the Canny and Hough transforms are used for edge detection and line detection, subsequently. Affected by light illumination, occlusions, and noises, the extracted straight lines contain lines that do not belong to a car's length direction. To reduce the interference of lines that have different directions with a car's length, we cluster the detected long straight lines into several classes according to their angles. The average angle of the cluster that has the most lines is taken as the main direction of the patch. Fig. 6 shows the results of patches after the rotation according to their main direction. As shown, our method effectively solves the rotation problem of the patches and rotates most of the cars to their vertical direction correctly.

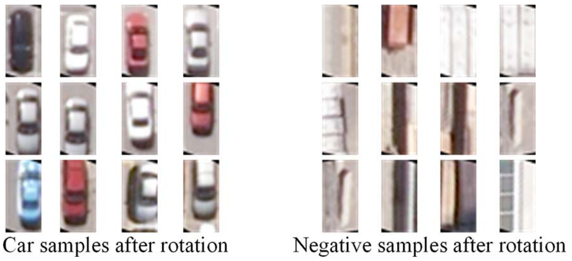


Fig. 6. Samples after rotating to their main directions.

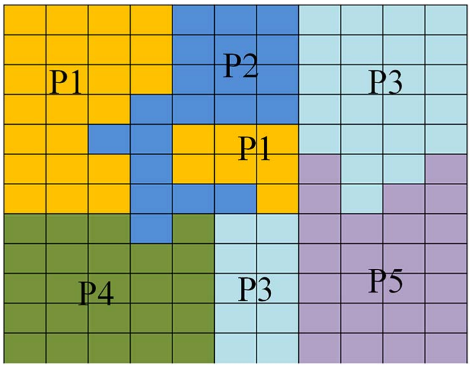


Fig. 7. Example of segmentation breakage. P1, P2, P3, P4, and P5 represent the partitions of an image. In the figure, P1 and P3 have two parts that are disconnected. The situations of P1 and P3 are the so-called segmentation breakage.

IV. Experiments and Discussions

In this part, we first provide a discussion regarding the superpixel segmentation. Then, we test our algorithm in two data sets, i.e., the Toronto data set and the OIRDS. In both data sets, the ground truths of test images were labeled with rectangular areas surrounding the cars. Only the detections that are located exactly on a ground truth are considered true detections. If one ground truth is redetected multiple times, only one is considered the true positive detection. Thus, other overlapping detections are considered as the false alarms.

A. Superpixel Segmentation Discussion

In this discussion, we tested our superpixel segmentation method on the public Berkeley database, which contains 300 images [54]. To analyze the robustness of our method to breakage, which is defined as the disconnection of segmentations, we evaluated the breakage rate of our method on the Berkeley database. Fig. 7 shows an example of segmentation breakage. In the figure, an image is segmented into five parts, namely, P1, P2, P3, P4, and P5. However, partitions P1 and P3 have two disconnected parts, denoted as segmentation breakage. Given segmentation P, the breakage rate is defined as

$$BR = \frac{\varphi(P)}{|P|} \tag{17}$$

where $\varphi(P)$ represents the number of disconnected segmentations of P and $|P|$ represents the number of total segmentations of P.

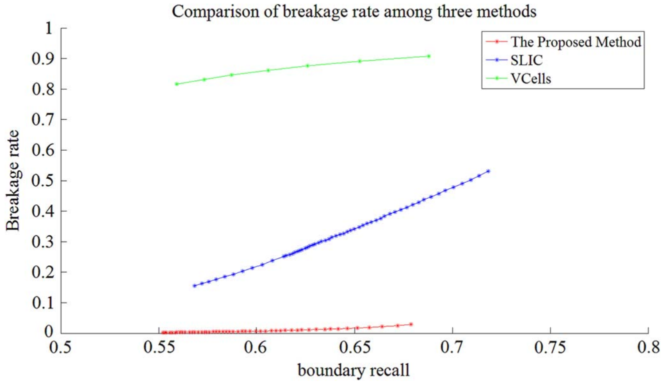


Fig. 8. Breakage rate comparison of segmentation results by our method, VCells, and SLIC in the Berkeley database. The red, blue, and green lines represent our method, SLIC, and VCells, respectively.

For comparison, we also evaluated the breakage rate of VCells [34] and SLIC [33] on this database. The code of VCells was obtained from the author, and the SLIC code was obtained from VLFeat. In our experiment, we fixed the superpixel size at approximately 200 pixels per superpixel and maintained a boundary recall rate ranging from 55% to 70%. Only the segmented boundary pixel, which is located on the boundaries of the ground truth, was considered as correct segmented boundary pixel. Fig. 8 shows the experimental result; the horizontal and vertical axes represent the boundary recall rate and breakage rate, respectively. In the figure, the red line represents our method, which shows that our method obtained the lowest breakage rate among the three methods. Both VCells and SLIC obtained high breakage rates when the boundary recall was higher than 0.55. The situation was worse for VCells. In contrast, during segmentation, we used the local information to improve the segmentation process and successfully controlled the segmentation with a low breakage rate. Fig. 9 shows a visual comparison of segmentation on an aerial image by our method, VCells, and SLIC. The segmentation parameters of each method were kept the same with the experiments in the Berkeley database. It can be observed that our method's segmentation result is smoother and more regular than that of the other two methods. When handling objects with complex textures (e.g., cars), the segmentation results of VCells and SLIC showed their breakages, which resulted in the generation of considerably small segmentation fragments. This increases not only the burden on the detection work but also the false alarm rate of detection.

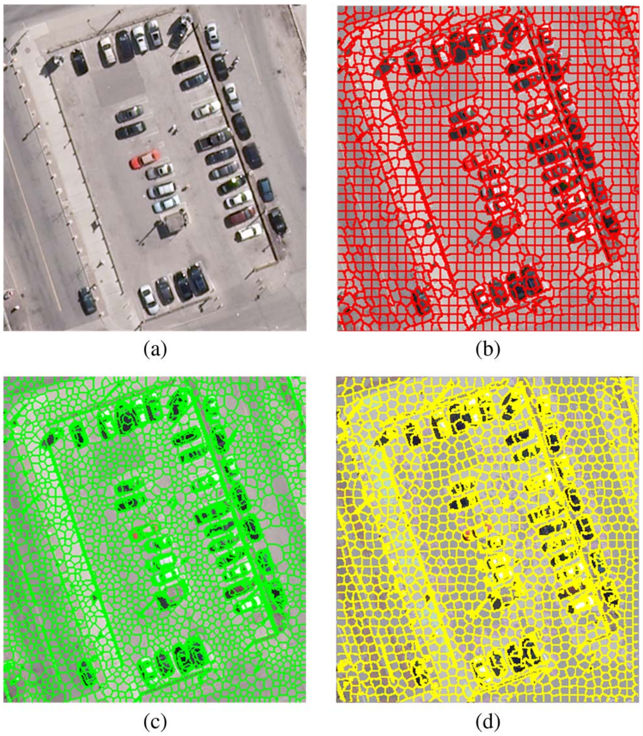


Fig. 9. Comparison of superpixel segmentation results among the proposed method, VCells, and SLIC. (a) Original high-resolution aerial image. (b)–(d) Segmentation results of the proposed method, VCells, and SLIC, respectively. The parameter configurations of each method were the same as those in the Berkeley database.

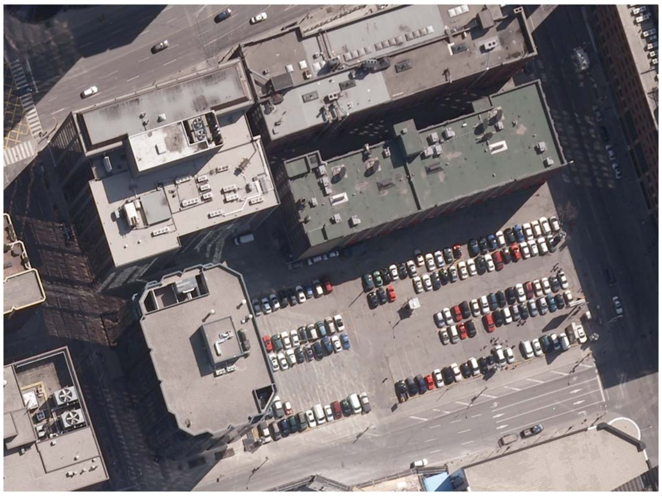


Fig. 10. Parking lot example from the Toronto data set.

B. Toronto Data Set

We tested the performance of our algorithm on an aerial image, covering the city of Toronto, with a size of 11 500 pixels × 7500 pixels and a color depth of 24 bits/pixel (RGB). The spatial resolution of the aerial image is 0.15 m, under which resolution, a car contains about 38 pixels × 16 pixels. In our experiment, we cut the image into subareas and selected several subareas for training and testing. Fig. 10 shows a subimage covering a parking lot. In the experiment, 13 subimages for training and 8 images for testing were selected. The total number of cars in the testing set is 1589. Generally, the scanning patch size is set at a size larger than the size of cars in the test images. In our experiments, the scanning patch size is 41 pixels × 21 pixels. The reason for using this scanning patch size is that we only consider the texture feature of vehicles in our method. The context information of nearby background would be studied in our future work.

TABLE I
SIZE OF THE SUBAREAS FOR TRAINING IN TORONTO DATA SET

Area	1	2	3	4	5	6	7	8	9	10	11	12	13
Width	1193	1758	1961	1629	1087	1374	1887	2086	1938	1523	2744	4060	1185
Length	775	1002	1136	1085	926	920	633	986	743	769	1264	2948	888

C. *Compact and Complete Training Set Construction*

In our experiment, we selected the positive patches (cars) and negative patches (the background) from 13 subareas for training a sparse representation dictionary. Table I shows the size of each subarea for training. We segmented all the subareas into superpixels with a size of approximately 400 pixels and generated 184 710 superpixels. Accordingly, 184 710 training patches, which include 5169 car patches, were obtained based on the superpixel centers. Choosing a small and complete training subset is necessary because the generated entire training set is too large for training and a large amount of information is redundant. However, it is difficult to manually select a compact and complete training subset. To automatically construct a complete training subset, we apply our training sample selection procedure to select representative training samples. In order to reduce the computational complexity, all the patches were first rotated to their main directions. Then, each patch was trimmed with a uniform size of 41 pixels × 21 pixels, as shown in Fig. 6.

From the rotated and trimmed training patches, we manually selected 60 car patches and 120 background patches, respectively. For the positives, we selected the car patches with clear textures without interference (shadows, occlusions, etc.). For the negatives, we selected the samples that appear similar to cars. The grids of HOG features were extracted as the initial sparse representation dictionary input. All positive features are labeled as 1, and all negative features are labeled as -1. Utilizing the trained dictionary, we calculated the sparse codes and classification scores of other remaining training patches. According to the calculated scores, some patches were selected to join the previously selected training subset. With the new selected training subset, a new dictionary was trained for the sample selection in the next iteration. We terminated the training sample selection after five iterations because, at that point, we obtained a classification accuracy value of greater than 80% under a 0.7 recall rate (considered a satisfactory detection accuracy value in this paper). Finally, a compact and complete dictionary training subset was created with 180 car patches and 1080 background patches.

Due to occlusions and illumination variations, the estimated main direction might not be a car's vertical direction of interest, resulting in omission detections of cars. Thus, we estimated two directions of each patch for scanning. After clustering the detected straight lines, we selected two clusters that contained more straight lines than others. The directions of the two selected clusters were defined as the main directions. A test patch was examined in the orientations of the two estimated main directions. To forbid redetections, only the result with a higher positive score was taken as the exam result. A test patch without lines was examined with no rotation in our experiment.

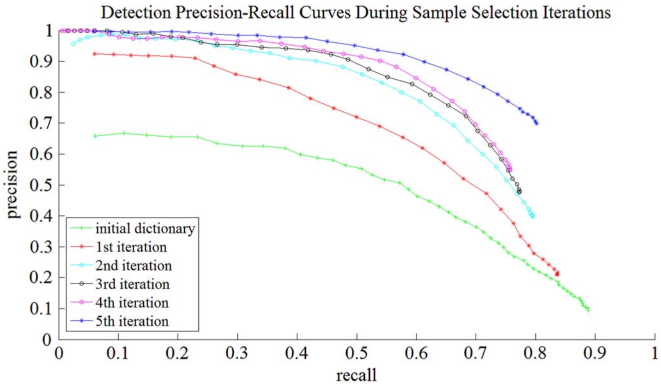


Fig. 11. Detection results with dictionaries trained in each iteration. Green, red, cyan, black, magenta, and blue represent the results of iterations ranging from initial to fifth, respectively.

To demonstrate the effectiveness of our iterative training sample selection, we used the dictionaries trained during iterations for vehicle detection on test images. Fig. 11 shows the detection precision–recall curves using dictionaries trained with samples selected in different iterations, ranging from the initial time to fifth time. The detection accuracy monotonously increases during the sample selection iterations because the negatives that have high similarity to positives are added into the training set one by one. With the addition of negatives that have a high similarity to positives, the decision boundary between the positives and negatives is more and more exact. Thus, the ability to separate the vehicles, as well as the negative tests that are close to vehicles in feature space, is increased during the iterations. Then, the detection accuracy improves during the iterations. The experimental result proves the effectiveness of our method for automatically constructing a compact and complete training set. In our experiment, we obtained a satisfactory result after five time iterations. In the following comparison, the fifth iteration result is regarded as our final result.

D. *Sensitivity of the Superpixel Size*

Superpixel segmentation is a vital process to detect cars from high-resolution aerial images in our method. The superpixel size influences the detection accuracy and recall rate. A large superpixel size reduces detection positions and false alarms, but it increases omission detections. On the contrary, a small superpixel size increases detection burden and redetections, but it improves recall rate. This is a tradeoff of the superpixel size for detection.

In this paper, to determine the best superpixel size, we tested five superpixel sizes (100, 225, 400, 625, and 900) to segment images for training and detection. From a segmentation

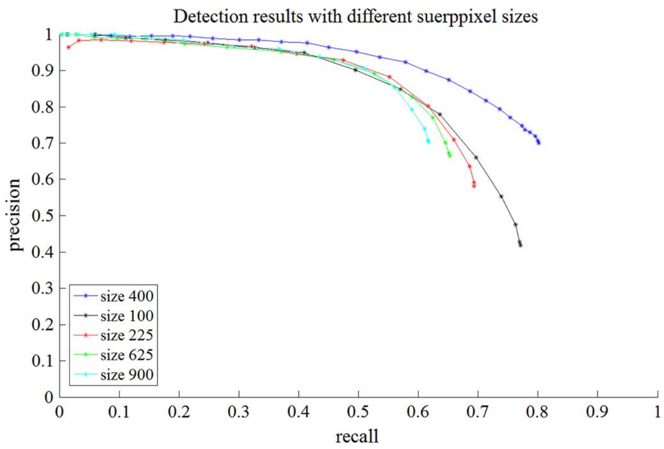


Fig. 12. Detection results of the proposed framework with different superpixel segmentation sizes. The superpixel sizes include 100, 225, 400, 625, and 900.

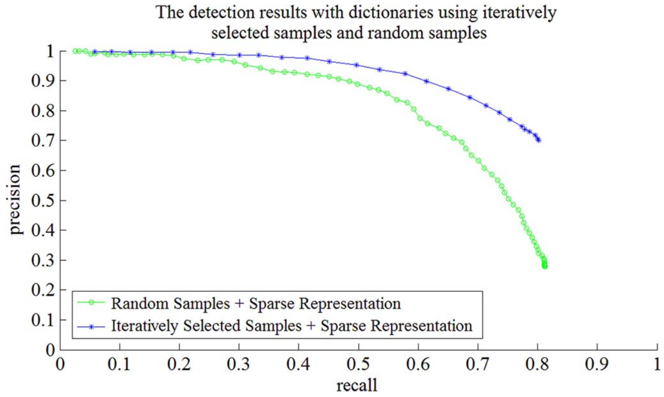


Fig. 13. Comparison of the detection results by using the iteratively trained dictionary and the randomly trained dictionary. The blue and green lines represent the iteratively selected samples' result and the random samples' result, respectively.

having a given superpixel size, we selected training samples and detected the test images. Fig. 12 shows the detection results of our framework with different superpixel sizes. In Fig. 12, the detection result with a superpixel size of 400 is superior to others. The detection accuracy quickly decreases with the superpixel sizes 100 and 225 because cars in the test images were oversegmented, which increased redetections during detection. On the contrary, the omissions increased when the superpixel size is 625 or 900, leading to a low maximum recall rate. To balance redetections and omission detections, the superpixel size was set at 400 in our study.

E. Effects on Training Samples

To analyze the influence of training samples on the accuracy of vehicle detection, we detected cars by our iteratively trained dictionary and a randomly trained dictionary, respectively. Each dictionary contained 180 cars and 1080 non-car objects. Fig. 13 shows the comparison results. The blue and green lines represent the precision–recall curves of detection results with our iteratively trained dictionary and the randomly trained dictionary, respectively. Both detections were under the proposed vehicle detection framework. As shown in Fig. 13, the iteratively trained dictionary method outperforms the randomly trained dictionary method for car detection. The method using the iteratively trained dictionary has higher precision for every recall rate value, which proves the high effectiveness of our iterative training sample selection strategy. The randomly selected training set was not able to effectively contain all the representative negatives in the original training set to train a highly discriminative dictionary.

F. Performance on Toronto Data Set

The Toronto data set was used to further verify the performance of our algorithm. Table II shows the sizes of the test images. In this paper, the test images have the same spatial resolution as the training images. We segmented the eight test images into superpixels with a size of 400 and generated patch candidates with a size of 61 pixels × 61 pixels based on the superpixel centers. These patches were rotated to their main directions and clipped to smaller patches with a size of 41 pixels × 21 pixels. We consider only the vehicle texture for the proposed method in this paper. The use of background information will be studied in our next work. During the detection, the sparse codes of patch candidates were calculated to classify them into cars or the background class. The dictionary in our experiment was obtained after five time training sample selection iterations. Fig. 18 shows two detection results of our method. In Fig. 18, the red line represents the wrong detections, and the green line represents the right detections. Our results show good performance, which has high vehicle detection recall and precision, in complex urban areas.

We also performed three other popular methods on the test images for comparison, including HOG + linear SVM, HOG + kernel SVM, and SIFT + linear SVM. All of the codes in our experiments were obtained from the publicly available sources (VLFeat). In these methods, a slide window scanning strategy with a slide step of 5 pixels on both the horizontal and vertical axes was used. For each scanning position, the test patch was rotated with a rotation interval step of 5° for examination. In the training stage, 180 car patches were selected and aligned vertically. Meanwhile, 1080 background patches were randomly selected as the negatives for training. The patch size for these methods was 61 pixels × 31 pixels, considering that a bit of background information benefits the performance of these methods in normal circumstance [55].

Fig. 14 shows the performance comparison between our method and the other three methods in the Toronto data set. The horizontal axis is the recall rate for vehicles, and the vertical axis represents the precision. The blue line represents the result of our method, which obtained higher precision than the other methods. The SIFT feature method had the worst performance among the four methods in our experiment. When the recall rate is higher than 0.6, our method still maintained high precision. However, the other three methods' precision values showed a dramatic decrease when the recall rate is higher than 0.6. Fig. 14 fully illustrates the better performance of our method.

TABLE II
SIZE OF THE SUBAREAS FOR DETECTION IN TORONTO DATA SET

Area	1	2	3	4	5	6	7	8
Width	1644	1382	1709	2605	1537	4357	4141	2280
Length	1152	864	1358	1002	1244	1770	1932	2042

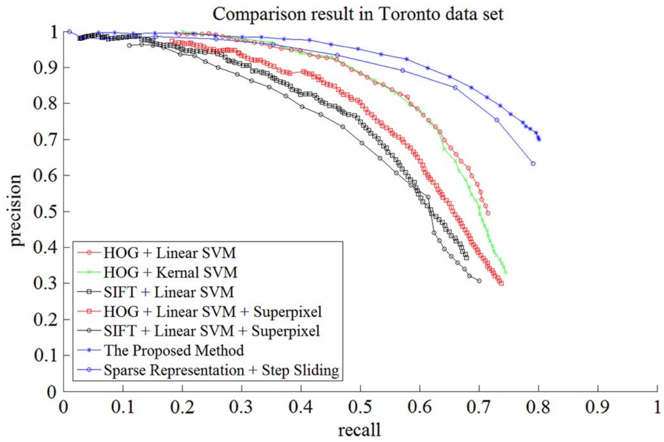


Fig. 14. Comparison between our method and the other methods in the Toronto data set.

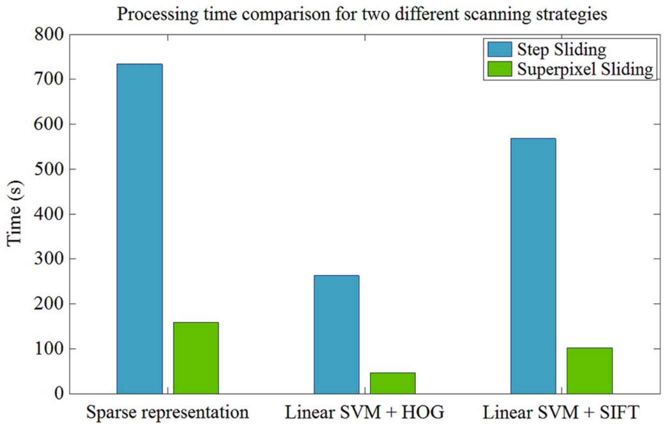


Fig. 15. Processing time comparison for two different scanning strategies, i.e., the scanning strategy with a fixed step sliding and the scanning strategy based on superpixel centers.

To examine the effects of detection scanning strategy based on superpixel centers, we also tested the performance of superpixel sliding strategy for linear SVM + HOG and linear SVM + SIFT. The test result for linear SVM + HOG + superpixel is shown as a red line with squares, and the test result for linear SVM + SIFT + superpixel is shown as a black line with circles in Fig. 14. The detection results show that the sliding strategy based on superpixel centers has little effect on the detection accuracy of linear SVM + HOG and linear SVM + SIFT, but the detection efficiency has been greatly improved, as shown in Fig. 15. In addition, we tested the sparse representation with fixed step sliding scanning strategy on this data set (see the blue line with circles in Fig. 14). The result shows that our proposed method has even a better performance than the method combining sparse representation with fixed step sliding strategy.

Fig. 15 shows a comparison of the processing efficiency. We conducted experiments on a personal computer with Intel Core i5-2400 CPU at 3.1 GHz and 8-GB RAM. The platform was the MATLAB, and the test was an image with the size of 352 × 379. In Fig. 15, blue bars represent the detection processing times using the traditional fixed step sliding strategy. The green bars represent the detection processing times using scanning strategy based on superpixel centers. As shown in Fig. 15, all the detections with fixed step sliding strategy consume much more time than their corresponding detections using scanning strategy based on superpixel centers. Among the detections using scanning strategy based on superpixel centers, our proposed method consumes a little more time than the other two methods. Nevertheless, our proposed method is still much more efficient than the detections with fixed step sliding strategy.

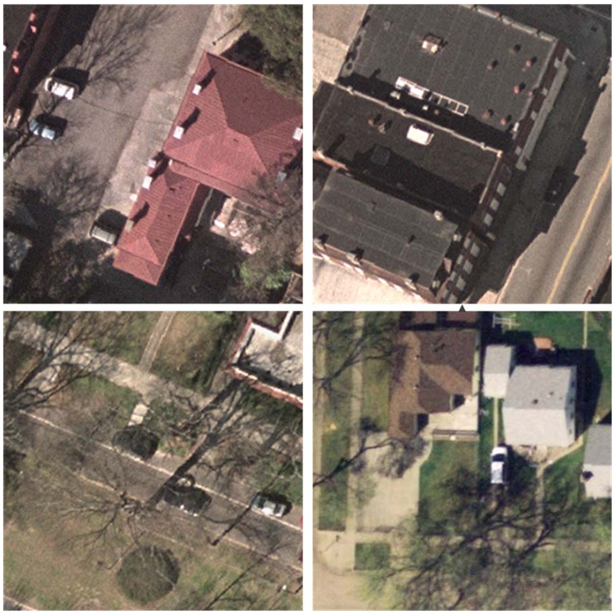


Fig. 16. Four test images in OIRDS.

G. Performance on OIRDS

To further verify the performance of our algorithm, the publicly available OIRDS, which contains 907 aerial images, was used. The total number of vehicles annotated in the data set is approximately 1800. Most images in this paper cover suburban areas, which leads to large number of cars that are partially or even totally occluded by trees, buildings, and other objects. Moreover, other factors such as spatial resolution and observation view variation also influence car detection negatively. In our experiment, to directly use the dictionary and SVM models trained in the previous experiment, images that have a spatial resolution different with 0.15 m × 0.15 m were manually eliminated.

Fig. 16 shows four selected test images in the OIRDS. In Fig. 16, most vehicles are occluded by trees, buildings,

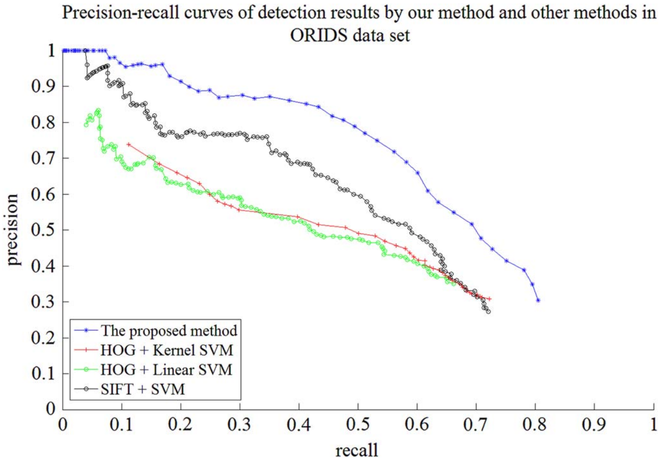


Fig. 17. Comparison of the detection results in OIRDS. The blue, red, and green lines represent the precision–recall curve of the proposed method, HOG + kernel SVM, and HOG + linear SVM, respectively.

and shadows cast by other elevated objects. In our study, 370 images containing 579 vehicles were selected to verify the performance of our vehicle detection method. Fig. 17 shows the comparative results of our method and the other three methods (i.e., HOG + linear SVM, HOG + kernel SVM, and SIFT + SVM). The blue line with stars represents the precision–recall curve of our method. The green, red, and black lines represent the precision–recall curves of HOG + linear, HOG + kernel SVM, and SIFT + SVM, respectively. In Fig. 17, our method's detection precision is higher than that of the other methods. When the recall rate is higher than 0.7, all the detection results of four methods are not satisfactory. Two reasons account for this phenomenon. First, the training set and the test set come from different data sets, leading to the differences of vehicles in the trained dictionary (or models) and the test images. The differences contain size, observation view, noise level, and illumination condition, which make it hard to achieve a high detection recall rate in the test images. Second, the occlusions, shadows, and brightness variations in OIRDS make it rather hard to detect the vehicles, resulting in a dramatic decrease in accuracy when we relax the threshold to detect those challenging vehicles with a high recall rate.

V. Conclusion

We have presented a novel vehicle detection method from high-resolution aerial images by using sparse representation and superpixel segmentation.

Through superpixel segmentation, aerial images are first segmented into superpixels. Based on the superpixel centers, we got meaningful patches for training and detection, benefiting the training sample selection and making the detection scanning highly effective. To construct a compact and complete training subset, we propose a training sample selection method based on sparse representation to select the most representative samples from the entire large training set. With the selected training subset, we obtain a sparse representation dictionary with highly discriminative ability for vehicle detection. We further improve the algorithm's effectiveness by using an effective direction estimation to make the patches maintain consistent directions during training and detection (see Fig. 18).

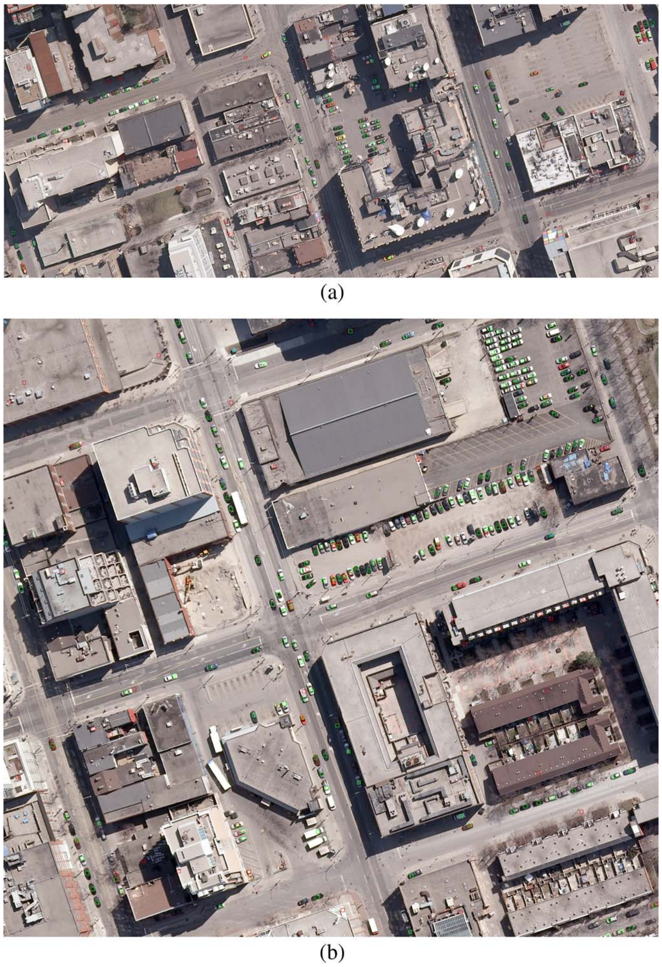


Fig. 18. Vehicle detection results of the proposed method in two subareas of the test image. In both (a) and (b), the locations with rectangles are the areas recognized as cars. The rectangles with a red color represent the wrong detection, and the rectangles with a green color represent the right detections. The redetections are also considered as wrong detection.

We tested our algorithm in two data sets, i.e., the Toronto data set and the OIRDS. Several state-of-the-art methods (i.e., HOG + linear SVM, HOG + kernel SVM, and SIFT + SVM) are compared with our method. The comparisons of the detection results show that our method obtained a satisfactory detection result and performed better than the compared methods. Three factors influence the detection accuracy of our method, namely, superpixel segmentation size, sample selection iteration time, and completeness of the original entire training set. The experimental analyses regarding the iteration time of training sample selection procedure and the processing efficiency were also presented in our experiments.

Although we have introduced a superpixel-based scanning strategy into our method to improve the detection efficiency, the sparse representation still has higher computational complexity than the SVM methods with same scanning strategy. Thus, in our future work, we will study a hierarchical classification structure to further improve the detection efficiency and accuracy.

References

[1] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video processing techniques for traffic flow monitoring: A survey," in *Proc. 14th Int. IEEE ITSC*, 2011, pp. 1103–1108.

[2] K. Mandal *et al.*, "Road traffic congestion monitoring and measurement using active RFID and GSM technology," in *Proc. IEEE 14th ITSC*, 2011, pp. 1375–1379.

[3] R. Du *et al.*, "Effective urban traffic monitoring by vehicular sensor networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 1, pp. 273–286, Jan. 2014.

[4] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 108–118, Jul. 2000.

[5] W. Liu, F. Yamazaki, and T. T. Vu, "Automated vehicle extraction and speed determination from QuickBird satellite images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 1, pp. 75–82, Mar. 2011.

[6] Z. Zheng, X. Wang, G. Zhou, and L. Jiang, "Vehicle detection based on morphology from highway aerial images," in *Proc. IEEE IGARSS*, 2012, pp. 5997–6000.

[7] J. Leitloff, S. Hinz, and U. Stilla, "Vehicle detection in very high resolution satellite images of city areas," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2795–2806, Jul. 2010.

[8] R. Ruskoné, L. Guigues, S. Airault, and O. Jamet, "Vehicle detection on aerial images: A structural approach," in *Proc. Int. Conf. Pattern Recog.*, 1996, pp. 900–900.

[9] X. Jin, and C. H. Davis, "Vehicle detection from high-resolution satellite imagery using morphological shared-weight neural networks," *Image Vis. Comput.*, vol. 25, no. 9, pp. 1422–1431, Sep. 2007.

[10] B. Salehi, Y. Zhang, and M. Zhong, "Automatic moving vehicles information extraction from single-pass WorldView-2 imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 1, pp. 135–145, Feb. 2012.

[11] A. Kembhavi, D. Harwood, and L. S. Davis, "Vehicle detection using partial least squares," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1250–1265, Jun. 2011.

[12] H. Grabner, T. T. Nguyen, B. Gruber, and H. Bischof, "On-line boosting-based car detection from aerial images," *ISPRS J. Photogramm. Remote Sens.*, vol. 63, no. 3, pp. 382–396, May 2008.

[13] T. Moranduzzo and F. Melgani, "Detecting cars in UAV images with a catalog-based approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6356–6367, Oct. 2014.

[14] Z. Zheng *et al.*, "A novel vehicle detection method with high resolution highway aerial image," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 6, pp. 2338–2343, Dec. 2013.

[15] X. Cao, C. Wu, P. Yan, and X. Li, "Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos," in *Proc. 18th IEEE ICIP*, pp. 2421–2424, 2011.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. CVPR*, 2005, pp. 886–893.

[17] T. Moranduzzo and F. Melgani, "A SIFT-SVM method for detecting cars in UAV images," in *Proc. IEEE IGARSS*, 2012, pp. 6868–6871.

[18] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. IEEE Conf. CVPR*, 2008, pp. 1–8.

[19] J. Wright *et al.*, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[21] A. Levinshtein *et al.*, "TurboPixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[22] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. IEEE ICCV*, 2011, pp. 1323–1330.

[23] X. Li, T. Jia, and H. Zhang, "Expression-insensitive 3D face recognition using sparse representation," in *Proc. Conf. CVPR*, 2009, pp. 2575–2582.

[24] F. Chen, H. Yu, and R. Hu, "Shape sparse representation for joint object classification and segmentation," *IEEE Trans. Image Process.*, vol. 22, pp. 992–1004, Mar. 2013.

[25] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[26] J. Zepeda, C. Guillemot, and E. Kijak, "Image compression using sparse representations and the iteration-tuned and aligned dictionary," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 1061–1073, Sep. 2011.

[27] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. CVPR*, 2008, pp. 1–8.

[28] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. CVPR*, 2009, pp. 1794–1801.

[29] J. Yang and M.-H. Yang, "Top-down visual saliency via joint CRF and dictionary learning," in *Proc. IEEE Conf. CVPR*, 2012, pp. 2296–2303.

[30] M. Cheng, C. Wang, and J. Li, "Sparse representation based pansharpening using trained dictionary," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 293–297, Jan. 2013.

[31] N. Yokoya and A. Iwasaki, "Object localization based on sparse representation for remote sensing imagery," in *Proc. IEEE IGARSS*, 2014, pp. 2293–2296.

[32] A. P. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Proc. IEEE Conf. CVPR*, 2008, pp. 1–8.

[33] R. Achanta *et al.*, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[34] J. Wang and X. Wang, "VCells: Simple and efficient superpixels using edge-weighted centroidal Voronoi tessellations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1241–1247, Jun. 2012.

[35] M.-Y. Liu, R. Chellappa, O. Tuzel, and S. Ramalingam, "Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 99–112, Jan. 2013.

[36] S. Hinz and A. Baumgartner, "Vehicle detection in aerial images using generic features, grouping, and context," in *Proc. Pattern Recognit.*, 2001, pp. 45–52.

[37] H. Moon, R. Chellappa, and A. Rosenfeld, "Optimal edge-based shape detection," *IEEE Trans. Image Process.*, vol. 11, no. 11, pp. 1209–1227, Nov. 2002.

[38] S. Hinz, "Detection and counting of cars in aerial images," in *Proc. ICIP*, 2003, vol. 2, pp. 997–1000.

[39] P. Reinartz, M. Lachaise, E. Schmeer, T. Krauss, and H. Runge, "Traffic monitoring with serial images from airborne cameras," *ISPRS J. Photogramm. Remote Sens.*, vol. 61, no. 3/4, pp. 149–158, Dec. 2006.

[40] J.-Y. Choi and Y.-K. Yang, "Vehicle detection from aerial images using local shape information," *Adv. Image Video Technol.*, vol. 5414, pp. 227–236, 2009.

[41] S. M. Khan, H. Cheng, D. Matthies, and H. Sawhney, "3D model based vehicle classification in aerial imagery," in *Proc. IEEE Conf. CVPR*, 2010, pp. 1681–1687.

[42] J. Xiao, H. Cheng, H. Sawhney, and F. Han, "Vehicle detection and tracking in wide field-of-view aerial video," in *Proc. IEEE Conf. CVPR*, 2010, pp. 679–684.

[43] H.-Y. Cheng, C.-C. Weng, and Y.-Y. Chen, "Vehicle detection in aerial surveillance using dynamic Bayesian networks," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2152–2159, Apr. 2012.

[44] T. Moranduzzo and F. Melgani, "Automatic car counting method for unmanned aerial vehicle images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 3, pp. 1635–1647, Mar. 2014.

[45] W. Shao, W. Yang, G. Liu, and J. Liu, "Car detection from high-resolution aerial imagery using multiple features," in *Proc. IEEE IGARSS*, 2012, pp. 4379–4382.

[46] H. Wang *et al.*, "Object detection in terrestrial laser scanning point clouds based on Hough forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1807–1811, Oct. 2014.

[47] H. Moon, R. Chellappa, and A. Rosenfeld, "Performance analysis of a simple vehicle detection algorithm," *Image Vis. Comput.*, vol. 20, no. 1, pp. 1–13, Jan. 2002.

[48] X. Zhou, W. Jiang, Y. Tian, and Y. Shi, "Kernel subclass convex hull sample selection method for SVM on face recognition," *Neurocomputing*, vol. 73, no. 10/12, pp. 2234–2246, Jun. 2010.

[49] F. Nie, H. Wang, H. Huang, and C. Ding, "Early active learning via robust representation and structured sparsity," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1572–1578.

[50] Z. Jiang, Z. Lin, and L. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[51] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal," *Commun. Pure Appl. Math.*, vol. 59, no. 6, pp. 797–829, Jun. 2006.

[52] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[53] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[54] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE ICCV*, 2001, pp. 416–423.

[55] G. Chen, Y. Ding, J. Xiao, and T. X. Han, "Detection evolution with multi-order contextual co-occurrence," in *Proc. IEEE Conf. CVPR*, 2013, pp. 1798–1805.

Ziyi Chen received the B.S. degree in computer science in 2010 from Xiamen University, Xiamen, China, where he is currently working toward the Ph.D. degree in the Department of Communication Engineering.

His current research interests include computer vision, machine learning, and remote sensing image processing.

Cheng Wang (M'11) received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2002.

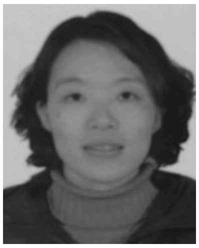
He is currently a Professor with and the Associate Dean of the School of Information Science and Technology, Xiamen University, Xiamen, China. He has authored more than 80 papers. His research interests include remote sensing image processing, mobile LiDAR data analysis, and multisensor fusion.

Prof. Wang is a member of SPIE and the IEEE Geoscience and Remote Sensing Society and a Council Member of China Society of Image and Graphics. He is the Cochair of ISPRS WG I/3.

Chenglu Wen (M'14) received the Ph.D. degree in mechanical engineering from China Agricultural University, Beijing, China, in 2009.

She is currently an Assistant Professor with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, China. She has coauthored more than 30 research papers published in refereed journals and proceedings. Her current research interests are machine vision, machine learning, and point cloud data processing.

Dr. Wen is the Secretary of the ISPRS WG I/3 on Multi-Platform Multi-Sensor System Calibration (2012–2016).



Xiuhua Teng received the B.Sc. degree in software theory from Fuzhou University, Fuzhou, China, in 2006.

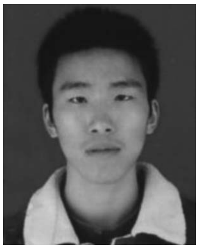
She is currently an Assistant Professor with the School of Information Science and Engineering, Fujian University of Technology, Fuzhou. Her current research interests include computer vision and machine learning.

Yiping Chen received the Ph.D. degree in information communication engineering from the National University of Defense Technology, Changsha, China, in 2011.

She is currently an Assistant Professor with the National University of Defense Technology and a Postdoctoral Fellow with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, China. Her current research interests include image processing, mobile laser scanning data analysis, and 3-D point cloud detection.

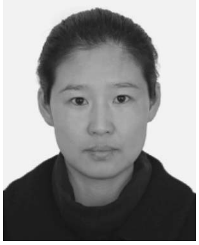
Haiyan Guan received the Ph.D. degree in geomatics from the University of Waterloo, Waterloo, ON, Canada, in 2014.

She is currently a Professor with the College of Geography and Remote Sensing, Nanjing University of Information Science and Technology, Nanjing, China. She has coauthored more than 30 research papers published in refereed journals, books, and proceedings. Her research interests include airborne, terrestrial, and mobile laser scanning data processing algorithms and 3-D spatial modeling and reconstruction of critical infrastructure and landscape.



Huan Luo received the B.Sc. degree in computer science from Nanchang University, Nanchang, China, in 2009. He is currently working toward the Ph.D. degree in Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, China.

His current research interests include computer vision, machine learning, and mobile LiDAR point cloud data processing.



Liujuan Cao received the Ph.D. degree in computer applied technology from Harbin Engineering University, Harbin, China, in 2013.

She is currently an Assistant Professor with the School of Information Science and Technology, Xiamen University, Xiamen, China. She has published more than 20 papers. Her current research interests include digital vector data security, large-scale image retrieval and remote sensing image processing.

Jonathan Li (M'00–SM'11) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa.

He is currently a Professor with the Key Laboratory of Underwater Acoustic Communication and Marine Information Technology of the Ministry of Education, School of Information Science and Engineering, Xiamen University, Xiamen, China. He is also a Professor with and the Head of the GeoSTARS Lab, Faculty of Environment, University of Waterloo, Waterloo, ON, Canada. He has coauthored more than 300 publications, more than 100 of which were published in refereed journals, including IEEE-TGRS, IEEE-TITS, IEEE-GRSL, ISPRS-JPRS, IJRS, PE&RS, and RSE. His current research interests include information extraction from mobile LiDAR point clouds and from Earth observation images.

Prof. Li is the Chair of the ISPRS WG I/Va on Mobile Scanning and Imaging Systems (2012–2016), the Vice Chair of the ICA Commission on Mapping from Remote Sensor Imagery (2011–2015), and the Vice Chair of the FIG Commission on Hydrography (2015–2018).