

Vehicle Detection and Motion Analysis in Low-Altitude Airborne Video Under Urban Environment

Xianbin Cao, *Senior Member, IEEE*, Changxia Wu, Jinhe Lan, Pingkun Yan, *Senior Member, IEEE*, and Xuelong Li, *Senior Member, IEEE*

Abstract—Visual surveillance from low-altitude airborne platforms plays a key role in urban traffic surveillance. Moving vehicle detection and motion analysis are very important for such a system. However, illumination variance, scene complexity, and platform motion make the tasks very challenging. In addition, the used algorithms have to be computationally efficient in order to be used on a real-time platform. To deal with these problems, a new framework for vehicle detection and motion analysis from low-altitude airborne videos is proposed. Our paper has two major contributions. First, to speed up feature extraction and to retain additional global features in different scales for higher classification accuracy, a boosting light and pyramid sampling histogram of oriented gradients feature extraction method is proposed. Second, to efficiently correlate vehicles across different frames for vehicle motion trajectories computation, a spatio-temporal appearance-related similarity measure is proposed. Compared to other representative existing methods, our experimental results showed that the proposed method is able to achieve better performance with higher detection rate, lower false positive rate, and faster detection speed.

Index Terms—Motion analysis, moving vehicle detection, spatio-temporal, urban environment.

I. INTRODUCTION

ONE OF THE goals of airborne moving vehicle detection systems is to obtain traffic statistics for urban planning

Manuscript received November 2, 2010; revised March 25, 2011; accepted June 26, 2011. Date of publication July 18, 2011; date of current version October 5, 2011. This work was supported in part by the National Basic Research Program of China (973 Program), under Grant 2011CB707000, in part by the National Natural Science Foundation of China, under Grants 61072093 and 60972103, in part by the Open Project Program of the State Key Laboratory of Computer Aided Design and Computer Graphics, Zhejiang University, under Grant A1116, in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China, under Grant 60921001, and in part by the Open Project Foundation of State Key Laboratory of Industrial Control Technology, Zhejiang University, under Grant ICT1105. This paper was recommended by Associate Editor A. Cavallaro.

X. Cao is with the School of Electronic and Information Engineering, Beihang University, Beijing 100083, China (e-mail: xbcao@buaa.edu.cn).

C. Wu and J. Lan are with the University of Science and Technology of China, Hefei 230026, China (e-mail: wcxia@mail.ustc.edu.cn; jhlan@mail.ustc.edu.cn).

P. Yan and X. Li are with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: pingkun.yan@opt.ac.cn; xuelong_li@opt.ac.cn).

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

Digital Object Identifier 10.1109/TCSVT.2011.2162274

and traffic jam relief. Compared with the stationary systems [1]–[3], [8], where cameras are usually mounted on street lamp posts or poles around traffic intersections, airborne systems are equipped on aerial aircrafts such as unmanned aerial vehicles [9] and helicopters [5], [6], having the advantages of low cost, high mobility, and wide view angle. At present, airborne moving vehicle detection has drawn a lot of interests from researchers [4], [10]. However, moving vehicle detection from low-altitude airborne platform, specifically under the urban environment, is a very challenging yet under-addressed problem. Some of the difficulties include cluttered background in urban environment, fast motion of the low-altitude airborne platform, and limited computing power. The detection of moving vehicles under urban environment in low-altitude airborne video is considered even more difficult, since it is expected to detect moving vehicles under the influence of buildings and other non-vehicle moving objects.

During the past two decades, some systems have been developed to detect vehicles in airborne platforms [6], [7], [11]–[13], [23]. Most of these systems are equipped with inertial measurement unit and high precision position and orientation system, which can provide precise state and position of the airborne platforms when the images are taken [12]. Nevertheless, the costs of those systems limit their uses in many practical applications.

In this paper, we propose an efficient machine learning-based method for vehicle detection and motion analysis in low-altitude airborne platform. First, to achieve high efficiency and simple computation, a boosting light and pyramid sampling histogram of oriented gradients (bLPS-HOG) feature extraction method is proposed to use together with a linear support vector machine (SVM) for vehicle detection. The output of vehicle detection is denoted by regions, which may potentially contain vehicles. Second, the above regions are refined for improving vehicle detection performance and computing the trajectories of vehicles for measuring traffic information. A spatio-temporal appearance-related similarity (STARS) measure is proposed for analyzing the motion of the detected vehicles. The STARS measure can help to effectively correlate vehicles from different frames and obtain the vehicle trajectories for analysis. Our experimental results demonstrated that compared with other representative existing algorithms, the proposed method can reach better performance

in terms of higher detection rate (DR), lower false positive rate (FPR), and faster detection speed.

The remainder of this paper is organized as follows. Section II provides a brief review of related works. In Section III, we present the proposed vehicle detection method using bLPS-HOG features and linear SVM. Section IV introduces the STARS measure and its application in vehicle motion analysis. The experimental results are provided in Section V, and, finally, this paper is concluded in Section VI.

II. RELATED WORK

A number of methods for detecting vehicles in low-altitude airborne platforms under urban environment have been developed [21], [22]. The existing methods can be roughly classified into two categories.

Methods in the first category are based on image processing techniques such as registration [14]–[16], optical flow [18], background subtraction [6], and stereo vision [17]. For example, Shastry and Schowengerdt [14] proposed a frame-by-frame video registration technique using Kanade–Lucas–Tomasi feature tracker [20] to automatically determine the correspondence between control points. The registration process maps video frames into a common coordinate system, thereby the airborne platform motion and altitude errors can be compensated and corrected, respectively. Ali *et al.* [33] performed moving objects detection by registering two consecutive frames followed by frame differencing. The spatial and motion relationship between objects is used to deal with the problem of missing detection and occlusion [37]. Reilly *et al.* [34] registered images using a point correspondence-based alignment algorithm and performed motion detection by building a background image model. However, due to the possibly large number of moving vehicles in each frame under the urban environment, the selected control points for registration can be easily obtained from moving vehicles. The frame-difference computed using such an incorrect registration may not be able to keep the integrity of the moving objects and result in poor detection performance. Yalcin *et al.* [18] proposed an approach based on optical flow to detect moving vehicles by segmenting the optical flow fields into background and occlusion layers. However, the complexity of urban environment may cause the optical flow fields to be very dense and lead to heavy computational load. Thus, it is not suitable for use on real-time platforms. Angel and Hickman [6] used the location, orientation, and scale information of vehicles predicted in a reference frame to align the adjacent frames. However, the assumption about the speed of vehicles may not be valid in urban environment and the computational load of their method is also intensive.

The second category includes machine learning-based detection methods, which have been becoming more and more popular recently [3], [19], [24]. In contrast to image processing techniques, these methods learn the patterns of vehicle appearance by using various kinds of features. Thus, they are generally less sensitive to image noise. For example, the work of Rosenbaum *et al.* [35] used gentle AdaBoost with Haar-like features to generate a confidence image, and obtained

regions of interest by clustering pixels based on the confidence image. SVM was used to further refine the detection results. However, the need of hardware support to align images limits their application. Zhao and Nevatia [19] combined several appearance features including the shadows of vehicles to detect both moving and static vehicles. A Bayesian network, which is trained by vehicle and non-vehicle samples, is setup to classify the objects. The boundaries of vehicle body, front windshield, and shadows are used as features. However, in airborne videos, these boundaries may become blurred and thus unreliable due to the platform motion. Cucchiara *et al.* [3] presented a traffic monitoring method by combining a pixel-based processing technique and a knowledge-based reasoning method, where occlusion reasoning helps to estimate the motion patterns and moving directions of vehicles in occluded regions effectively. The shortcoming of this method is that the knowledge is difficult to build, which leads to insufficient learning.

In this paper, we present a novel method for vehicle detection and motion analysis, which addresses the problems of low video quality, cluttered scenes, large illumination change, and fast platform motion. The main contributions of our paper include the new bLPS-HOG descriptor for using with linear SVM classifier and the STARS measure for motion analysis [38].

III. VEHICLE DETECTION

In this paper, a learning-based vehicle detection method is proposed. Two most important components of the method are classifier training and feature description. In our paper, SVM classifier is used for vehicle detection. The reason is that SVM is able to obtain the global optimal solution, while other learning methods, such as rule-based methods and neural networks, are often trapped by local minima [29]. Besides classifier training, another key component in vehicle detection is to obtain good feature description that is able to distinguish vehicles from other objects. Compared with features like scale-invariant feature transform [26], local binary feature [27], and Shapelet feature [28], histogram of oriented gradients (HOG) feature [25] shows better performance in characterizing object shape and appearance. In addition, HOG is not sensitive to illumination change. However, the high dimensionality of the HOG features will lead the training and classification using SVM with sophisticated kernel functions to dramatically slow down. It may also cause the model to be over fitted [29].

In our paper, a linear SVM classifier is employed, which can avoid over fitting and perform classification efficiently. To speed up vehicle detection and enhance the overall performance, a new bLPS-HOG feature extraction method is introduced to reduce the dimensionality of the input features for SVM. The flowchart of the proposed method using linear SVM classifier with bLPS-HOG features is shown in Fig. 1. The details of the feature extraction and classifier training are presented in the following sections.

A. Extracting bLPS-HOG Features

Due to their effectiveness, the HOG feature descriptor and the corresponding detection scheme proposed by Dalal and

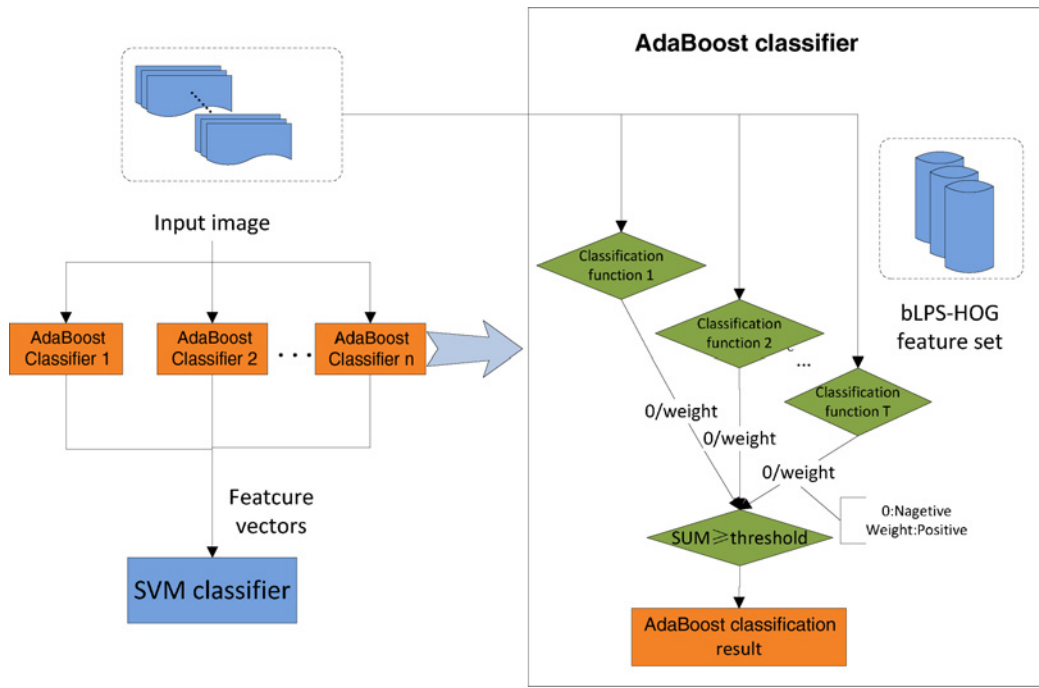


Fig. 1. Flow chart of the SVM classifier: input images are divided into blocks, each of which corresponds to a strong classifier trained by AdaBoost using LPS-HOG features, and all AdaBoost outputs are combined to establish the final SVM feature vector.

Triggs [25] have been widely used for pedestrian detection. The HOG features describe the distribution of image gradients in different orientations. It is often used to capture shape and appearance features for object detection. These descriptors are computed by dividing an image patch into smaller connected regions (cells). For each cell, a histogram of gradient orientations is computed. The overall HOG feature of the input image patch is obtained by combining all the cell histograms into one vector [25]. However, the high dimensionality of HOG features results in the increased computational cost. Wang *et al.* [30] and Jia and Zhang [36] tried to solve this problem by using the boosting scheme. However, the computational cost is not so satisfactory. To further speed up the detection process and better represent the global features of vehicles, in this paper, we propose to lightly sample the gradients and use a pyramid structure for computing HOG and boost the obtained features to reduce the dimensionality of the feature vectors. The new algorithm is coined as bLPS-HOG.

1) *Histogram of Oriented Gradients*: HOG descriptors are computed from image gradients and are designed to be discriminative, while remaining robust to small image changes. Overall, HOG feature is defined by the statistical information of the gradient orientation and intensity of a rectangular area. It is computed as follows.

First, the gradients of an input rectangular image patch are computed as

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y) \quad (1)$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1). \quad (2)$$

Here, $G_x(x, y)$ and $G_y(x, y)$ represent the horizontal and vertical gradients of the pixel point (x, y) , respectively. $H(x, y)$

refers to the gray value of the input image at (x, y) . Then the overall gradient intensity $G(x, y)$ and orientation $\alpha(x, y)$ are computed as

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (3)$$

$$\alpha(x, y) = \tan^{-1} \left(\frac{G_y(x, y)}{G_x(x, y)} \right). \quad (4)$$

After calculating the gradients, the orientations are divided into several bins. For each pixel point, the gradient intensity contributes to each orientation bin as

$$V_k(x, y) = \begin{cases} G(x, y), & \alpha(x, y) \in \text{bin}_k \\ 0, & \alpha(x, y) \notin \text{bin}_k \end{cases} \quad (5)$$

where $V_k(x, y)$ is the gradient intensity in the k th bin for pixel at point (x, y) .

Dalal and Triggs [25] extracted HOG feature using blocks with size of 16×16 and divided each block into four cells. In order to eliminate the impacts of lighting variation, the gradient histogram is normalized in each block as follows:

$$f(C_i, k) = \frac{\sum_{(x, y) \in C_i} V_k(x, y) + \varepsilon}{\sum_{(x, y) \in B} V_k(x, y) + \varepsilon} \quad (6)$$

where $f(C_i, k)$ represents the normalized gradient value of the k th bin in cell C_i contained in block B . ε is a very small number in order to avoid being divided by zero.

2) *Light Sampling*: When computing HOG features, 2×2 cells are often used in every block. With smaller cells, the extracted features would be more descriptive for the details of the object. However, such a division scheme may not be good for real-time vehicle detection. One reason is that the computational cost is high, since the number of cells is

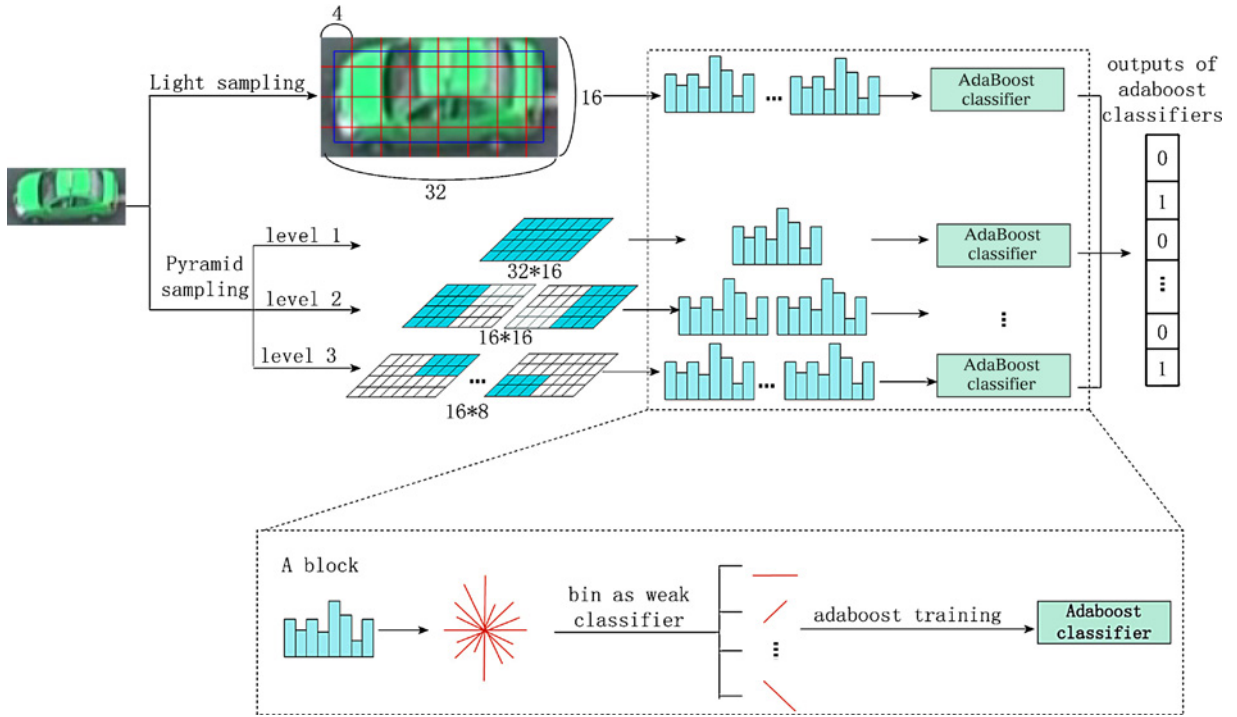


Fig. 2. Workflow for extracting the bLPS-HOG features.

large. The other reason is that the features can be sensitive to illumination changes and object shape variations because too much detail is presented. Therefore, we propose to use light sampling for feature extraction.

Instead of computing HOG in cells, light sampling is directly applied to blocks without cell division and the sizes of blocks are set to 4×4 uniformly to extract local features. The proposed light sampling technique has two stages. In the first stage, each input image is divided into small blocks with the size of 4×4 pixels. Then gradient is computed for each block. After that, a $8 \times 4 \times k$ -dimensional vector is obtained for the input patch, where k refers to the total number of bins. In the second stage, blocks with the same size are extracted from the central part of the image patch to obtain more local features. In this stage, different local information can be obtained, since these new blocks occupy different patch areas from those in the first stage. Given the size of our sample patch, the central part with the dimension of 28×12 is extracted as shown in Fig. 2 (the blue box in the top row for light sampling). Another vector of $7 \times 3 \times k$ dimension is obtained.

After the above two stage processing, the total dimensionality of the lightly sampled features becomes $53 \times k$. Since there is no cell division, the feature normalization is changed from the original operation in (6) to

$$f(k) = \frac{V_k(x, y) + \varepsilon}{\sum_{(x, y) \in B} V_k(x, y) + \varepsilon}. \quad (7)$$

By light sampling, blocks can be obtained from the central part area in addition to the entire image patch. Thus, plenty of local information from different areas can be extracted, which makes the feature more discriminative for the details.

3) *Pyramid Sampling*: Block size setting in light sampling reflects the local details. However, the global structural information is missed. With the purpose of extracting global features, pyramid structure is introduced to divide an image patch into blocks in different scales for histogram computation.

By using different block sizes, various features which are rich in global information can be obtained. Specifically, the block sizes have a pyramid structure changing from larger blocks to smaller ones. Suppose that the image size is $2^m \times 2^n$, the size of each block by pyramid sampling can be defined as $2^i \times 2^j$ ($i = 3, 4, \dots, m, j = 2, 3, \dots, n$). To avoid redundancy, all blocks are larger than those in light sampling. In our sample sets, the sample patches have the size of 32×16 (width \times height). A brief schematic explanation of pyramid sampling is shown in the middle row of Fig. 2.

By pyramid sampling, since block sizes are set to $2^i \times 2^j$, the total number of blocks is $(2^m / 2^i) \times (2^n / 2^j) = 2^{m+n-i-j}$. Thus, if there are k bins in each block, the total number of dimensions reaches

$$k \sum_{\substack{i=3 \dots m \\ j=2 \dots n}} 2^{m+n-i-j}.$$

Combined with the light sampling, the dimensionality of the complete LPS-HOG features is

$$k \sum_{\substack{i=3 \dots m \\ j=2 \dots n}} 2^{m+n-i-j} + 53 * k.$$

After LPS, in contrast to the original HOG feature, total dimensionality is significantly reduced due to two reasons: 1) cell division is not considered, and 2) blocks are non-overlapping. In addition, a lot of local and global information



Fig. 3. Some (a) positive and (b) negative samples used for training the classifiers.

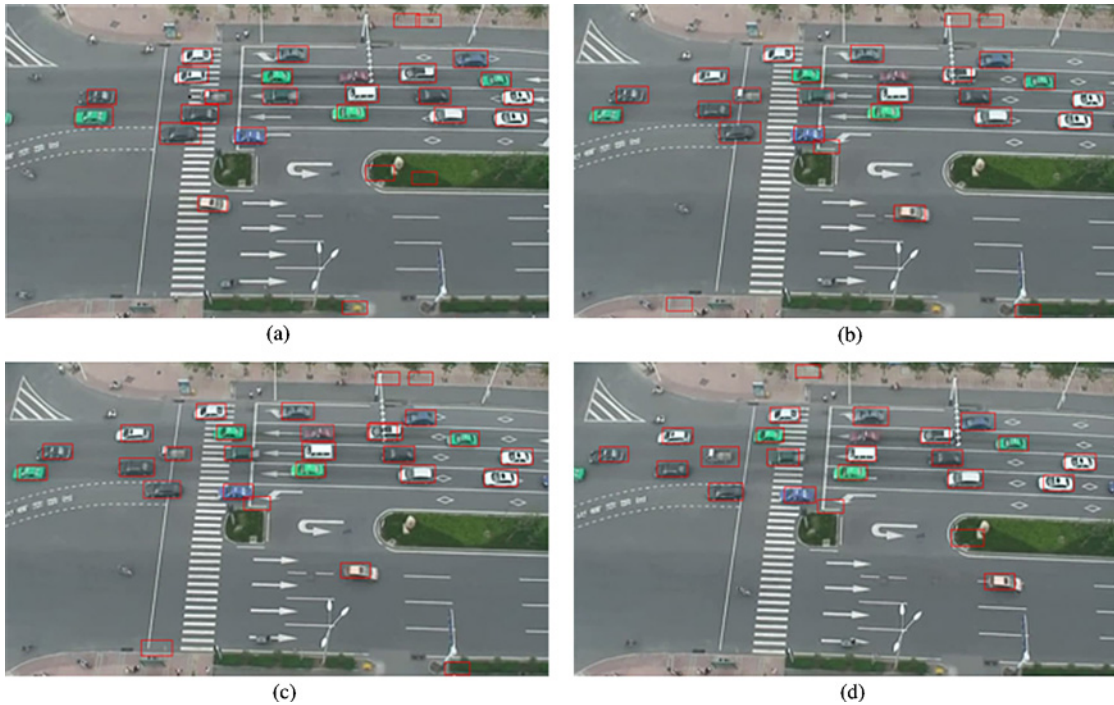


Fig. 4. Output of the SVM classification in different frames. (a) Detection results in frame t_0 . (b) Detection results in frame t_1 . (c) Detection results in frame t_2 . (d) Detection results in frame t_3 .

can be extracted so that LPS-HOG can reach comparable performance with the original HOG feature, with the advantage of much lower dimensionality.

However, if a large k value is given, the total dimensionality is still too high for a real-time vehicle detection system to handle by slowing down the speed of classifier during window sliding. In order to speed up the training and detection speed, the extracted LPS-HOG features are further boosted by using AdaBoost technique.

4) *bLPS-HOG Features*: In our proposed method, as the LPS-HOG feature is a histogram with k bins in every block, to further reduce the dimensionality, those bins are combined into a single unit. The idea of AdaBoost is applied by considering those bins as a set of weak classifiers. Let $H_N = \{b_1, \dots, b_N\}$ denote a LPS-HOG descriptor with N bins. By comparing each bin b_i with its corresponding threshold θ_i obtained during AdaBoost training, we can get the weak classifiers corresponding to the bins. Formally, let $h_{ji}(x)$ denote the i th weak classifier, which can be written as

$$h_{ji}(x) = \begin{cases} 1, & \text{if } p_i f_{ji}(x) < p_i \theta_{ji} \\ -1, & \text{otherwise} \end{cases} \quad (8)$$

where x represents an input image patch, $f_{ji}(x)$ denotes the value of the i th bin in the j th block, θ_{ji} is a threshold used to make a decision for $f_{ji}(x)$, and p_i is a polarity parameter used to change the direction of the inequality, which is either -1 or $+1$. These weak classifiers are then gathered together to form a strong classifier. The output of the corresponding AdaBoost classifier is used as an entry to establish a bLPS-HOG feature vector. An example of combining eight bins into one strong classifier is shown in Fig. 2.

B. Linear SVM Classifier

The output of the AdaBoost training on each block is a strong classifier. In the proposed method, the strong classifiers from all the blocks of an input image patch are used to construct a feature vector. A linear SVM classifier is then trained using the feature vectors for vehicle classification.

As it has been indicated by many machine learning methods [3], [19], [24], the number of high-quality samples plays a key role in classifier training. In our paper, to train the classifiers to get good performance, 2048 positive samples were manually selected and 100 000 negative samples were

generated automatically by programs using image frames without vehicles. All the samples were scaled to the size of 32×16 (width \times height) for training as shown in Fig. 3. The proposed bLPS-HOG features are designed to train a series of AdaBoost classifier to detect vehicles.

For simplicity, the final classifier is trained by vehicle samples that are only in horizontal orientation. To detect vehicles in different orientations, each sliding window can be rotated. In our experiments, the sliding window is rotated every 20° from 0 to 180° . That is, each sliding window is checked at nine different orientations to see whether it contains a vehicle or not. Some vehicle detection results obtained by the SVM classification are shown in Fig. 4, where the detected vehicles are marked with red rectangles. As we can see, most of the vehicles have been correctly detected. However, there exist some false positives, i.e., some non-vehicle objects were classified as vehicles. In addition, some vehicles were missed. Temporal information existing in the videos may be used to improve the performance of vehicle detection.

In our proposed method for vehicle detection, temporal information is not used in SVM classification. The most important reason is that the motion of moving platforms is usually unknown, which may lead the detection methods incorporated with motion to fail. In order to use the motion information to help vehicle detection in the classification stage, the motion of the platform itself has to be first computed from the images, which can be computationally expensive. In our paper, after the vehicles are detected, an efficient motion analysis step is proposed to correlate the detected vehicle windows. The proposed motion analysis method is not only able to analyze the motion of the moving vehicles but also able to refine the detection results. Another much desired property of the motion analysis method is that it has very low computational load, which makes it suitable for use on real-time applications. The details of our vehicle motion analysis method are presented in the next section.

IV. VEHICLE MOTION ANALYSIS

In our paper, since vehicles are detected in each frame independently, the vehicles have to be correlated together across the frames for motion analysis. In order to robustly map the vehicles correctly to their corresponding ones, a STARS measure is proposed to match the same vehicles across different frames. Some vehicle correlation results using the proposed STARS measure are shown in Fig. 5.

Since the motion information is not used in the detection process, it is possible that some vehicles may be missed in certain frames but are detected in their neighboring frames. In addition, some non-vehicle objects can be mistakenly detected as vehicles in some frames. However, our observation shows that most of these errors only appear in several discontinuous frames. Thus, our proposed STARS-based method uses a number of consecutive frames to establish the correlation of vehicles to refine the detection results for motion analysis. By enforcing the consistence of correspondences, the missed vehicles can be recovered and those non-vehicle objects may be removed. By using the refined results, vehicle trajectories can be computed.

For the ease of presentation, the outputs of the SVM classifier are named as detected windows and the i th detected window in frame t is denoted by W_t^i . For each detected window W_t^i in frame t , the STARS between W_t^i and each of its possibly matched detected window W_{t+1}^j in frame $t+1$ is computed. The one with the highest STARS score above a predefined threshold is considered as the detected window corresponding to the same vehicle of W_t^i . By using the STARS measure, the detected windows of the same vehicle in different frames can be matched. As shown in Fig. 4, after identifying the correspondences, the marked vehicles are tracked in the video sequence.

In the proposed method, STARS between two detected windows W_t^i and W_{t+1}^j uses both color and spatial information to match the detected windows as

$$S(W_t^i, W_{t+1}^j) = \rho_1 S_1(W_t^i, W_{t+1}^j) + \rho_2 S_2(W_t^i, W_{t+1}^j) \quad (9)$$

where $S(W_t^i, W_{t+1}^j)$ is the total STARS between two detected windows in frame t and $t+1$, $S_1(W_t^i, W_{t+1}^j)$, and $S_2(W_t^i, W_{t+1}^j)$ are the similarities based on color distribution and spatial locations, respectively, and ρ_1 and ρ_2 are two positive weight factors satisfying the constraint of $\rho_1 + \rho_2 = 1.0$.

A. Color Similarity

The color similarity $S_1(W_t^i, W_{t+1}^j)$ is measured by using the hue saturation value (HSV) color histogram [29], as it has been shown that HSV histogram may better represent the color information of an image. The total number of bins of the HSV histogram is $N = N_h N_s + N_v$, where N_h , N_s , and N_v are the numbers of bins of hue, saturation, and value components, respectively. The i th detected window with center c in frame t is represented by $W_t^i(c)$. The maximal distance from any point in the detected window to the window center is defined as

$$a = \frac{1}{2} \sqrt{h^2 + w^2}$$

where h and w denote the height and the width of the detected window, respectively. It is reasonable to assume that the pixels close to the centers of the detected windows are usually more important than those pixels far from the centers in computing the color similarity. Thus, when computing the color histogram of each detected window, we assign the weight

$$\omega(r) = \begin{cases} 1 - r^2, & r < 1 \\ 0, & \text{otherwise} \end{cases}$$

to each pixel, where $r = \frac{|u-c|}{a}$ is the distance between the pixel u and the center c of the detected window. The color histogram of the detected window $W_t^i(c)$ is then computed as follows:

$$p_t^{(n)}(c) = K \sum_{u \in R_t(c)} \omega\left(\frac{|u-c|}{a}\right) \bullet \delta(b_t(u) - n), \quad n = 1, \dots, N \quad (10)$$

where δ is the Kronecker delta function, $b_t(u)$ denotes the HSV index of pixel u , and the normalization factor $K =$

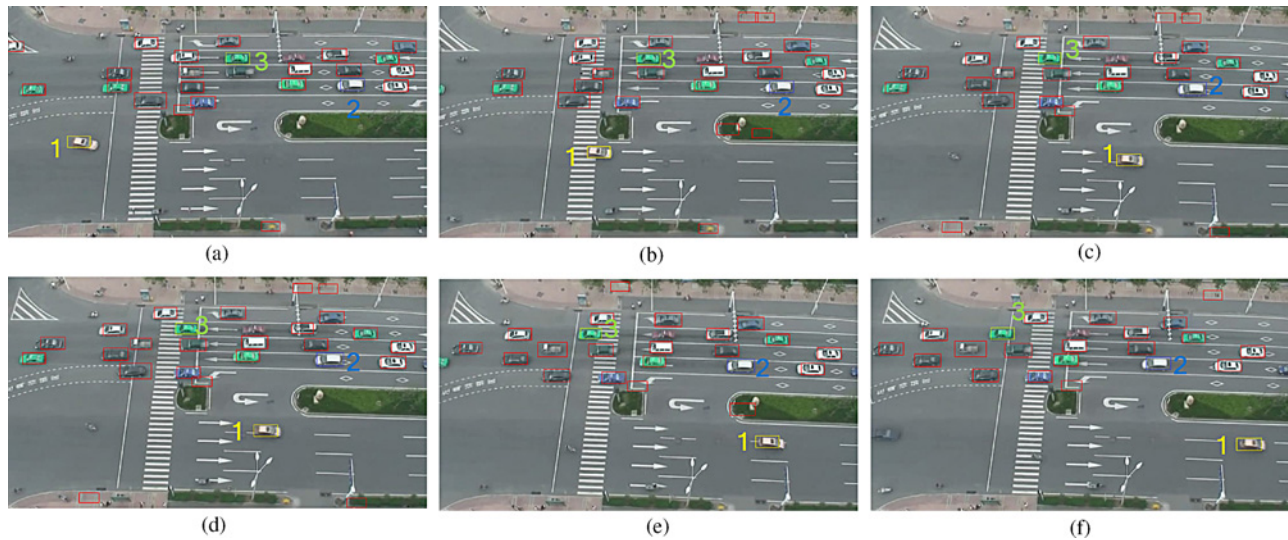


Fig. 5. Results of vehicle correlation across video frames using STARS measure. (a) Frame t_0 . (b) Frame t_1 . (c) Frame t_2 . (d) Frame t_3 . (e) Frame t_4 . (f) Frame t_5 .

$\frac{1}{\sum_{u \in R_t(c)} \omega(\frac{|u-c|}{a})}$ is used to ensure $\sum_{n=1}^N p_t^{(n)}(c) = 1$. The Bhattacharyya distance is adopted to measure the color similarity as

$$S_1(W_t^i, W_{t+1}^j) = \sum_{n=1}^N \sqrt{p_t^{(n)} p_{t+1}^{(n)}} \quad (11)$$

where $p_t(c_1)$ and $p_{t+1}(c_2)$ are the color histogram of $W_t^i(c_1)$ and $W_{t+1}^j(c_2)$, respectively. If $p_t(c_1)$ and $p_{t+1}(c_2)$ are from the same vehicle, the similarity $S_1(W_t^i, W_{t+1}^j)$ should be close to 1. Otherwise, it is supposed to be small.

B. Spatial Similarity

The other component of the STARS is the spatial position correlation. It is obvious that in airborne videos, detected windows corresponding to the same vehicle in neighboring frames are spatially close to each other, i.e., similar in window size and position. Thus, in order to evaluate the similarity of position and size of two detected window, the spatial similarity is measured by

$$S_2(W_t^i(c_1), W_{t+1}^j(c_2)) = \exp \left\{ -\frac{1}{2\alpha} \left[\frac{(x_{c_1} - x_{c_2})^2}{\sigma_x^2} + \frac{(y_{c_1} - y_{c_2})^2}{\sigma_y^2} + \lambda \left[\frac{(H_{c_1} - H_{c_2})^2}{\sigma_H^2} + \frac{(W_{c_1} - W_{c_2})^2}{\sigma_W^2} \right] \right] \right\} \quad (12)$$

where $c_1 = (x_{c_1}, y_{c_1})$ and $c_2 = (x_{c_2}, y_{c_2})$, H is the height of the detected window, W denotes the width, and λ is a constant factor to adjust the importance of the position and the size of the detected windows. The spatial similarity in (12) measures the similarity of both the position (x, y) and the size (H, W) between two detected windows.

The position constraint in (12) also limits the search range for window matching. It assumes that the vehicle movement is small between two continuous frames. This is usually the case for vehicles moving from one frame to its immediately next

frame. Therefore, in our experiments, we only consider three consecutive frames at times t , $t+1$, and $t+2$ in each group for detected window matching. To deal with the recovery of possibly missed vehicles and the removal of false positives, the algorithm will try to find the corresponding window in the frame after it at $t+2$, if the matched correspondence of a window in frame t cannot be found in the following frame at $t+1$. The times that a window W appearing in the three consecutive frames is counted. Ideally, it should be equal to 3, if the vehicle detection works perfectly. If it is 2, W is considered as a vehicle with missed detection in one of the frames. Then the missed window location will be estimated using the positions from the other two frames. Otherwise, it is considered that a false alarm happened in one of the frames. The corresponding detection window will be removed. Thus, the motion analysis step also brings us the benefits of decreasing the FPR and increasing the DR by using the STARS measurement. The processing steps are summarized using pseudo-code and given in Table I.

After the vehicles are matched between frames, the trajectories of the vehicles can be computed. As shown in Fig. 6, the trajectory of three vehicles marked through a sequence of frames is drawn in yellow curves. Fig. 7 shows the corresponding vehicle in a mosaic image after registering the frames together, where the numbers adjacent to each rectangle are the frame numbers in which the vehicle appeared at the corresponding position.

V. EXPERIMENTS AND DISCUSSION

In this section, experimental results of the proposed method on traffic videos are presented to show the performance of our approach.

A. Datasets and Experiment Platform

The experiments were performed on our videos in urban traffic environment and highway traffic scenes and the public DARPA VIVID datasets. Some snapshots of the videos with

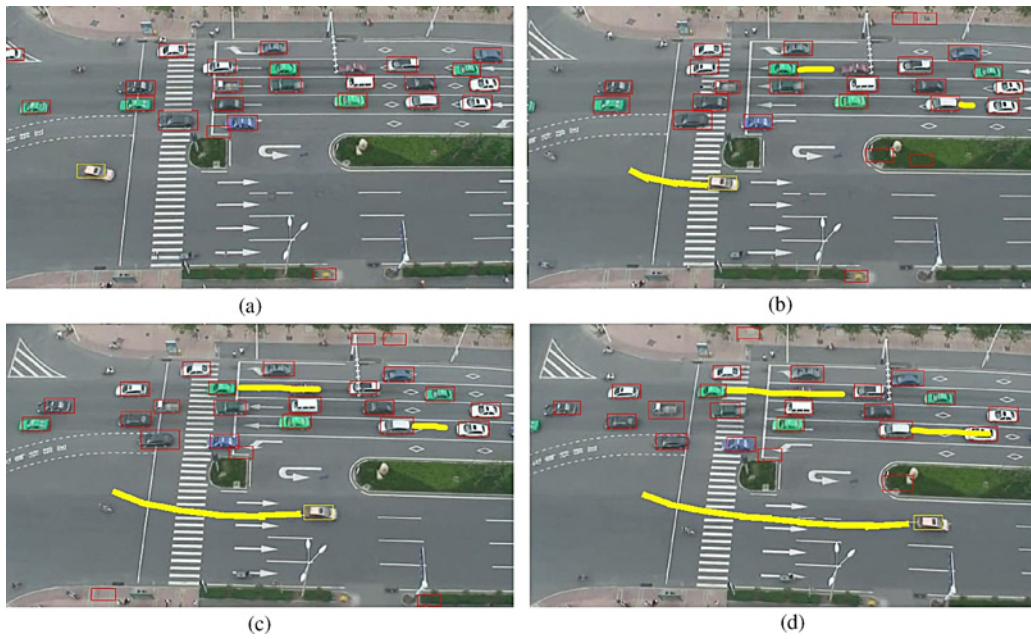


Fig. 6. Trajectory of moving vehicles through a sequence of frames. (a) Frame t_0 . (b) Frame t_1 . (c) Frame t_2 . (d) Frame t_3 .

TABLE I
VEHICLE CORRELATION BY STARS METRIC

Input: detected windows in three consecutive frames at time $t - 1, t, t + 1$
Output: vehicle labels in different frames
1) For each detected window W_t^i at frame t
a) find the detected window W_{t-1}^j in frame $t - 1$ with the highest similarity θ_1 to W_t^i by STARS metric in (9):
i) if $\theta_1 > \tau$, where τ is a predefined threshold,
the detected window in frame $t - 1$ is considered valid;
b) similar as a), find the detected window in frame $t + 1$ with the highest similarity θ_2 to W_t^i by STARS metric
i) if $\theta_2 > \tau$,
the detected window in frame $t + 1$ is considered valid;
c) if there is no valid detected window, W_t^i is considered as false positive;
d) else:
i) if either the detected window in frame $t - 1$ or the one in frame $t + 1$ is valid,
there is a vehicle missed in one of the frames. The missed window location at frame $t + 1$ will be estimated using the positions from the other two frames;
ii) the detected windows in all of the three frames are labeled as the same vehicle.

vehicle detection results are shown in Fig. 8. Our videos were captured at different scenes under different illumination and traffic conditions (crowded and sparse). Urban traffic videos were captured around the height of 60–90m using a digital video camera (Sony-DCR-HC21E) while the video of highway was captured by an unmanned aerial vehicle. Quantitative evaluation was performed on the video segments extracted from all the sequences of our data set and the DARPA VIVID datasets. The experiments were carried out on a computer with 3.2GHz CPU and 4GB double data rate RAM.

B. Performance Evaluation

The detection speed in terms of frames per second (f/s), the DR, and the FPR were used to quantitatively evaluate the performance of our method. The DR and FPR are defined

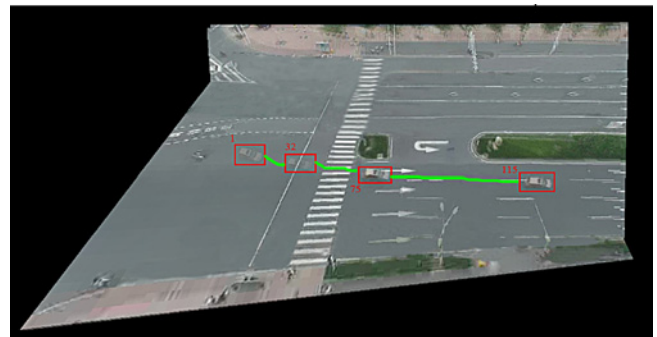


Fig. 7. Trajectory of a moving vehicle in the stitched frame after motion analysis.

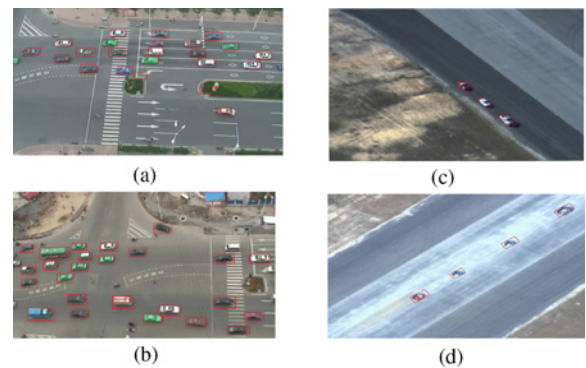


Fig. 8. (a), (b) Vehicle detection results in urban traffic scenes. (c), (d) Vehicle detection result in DARPA VIVID videos.

as

$$DR = \frac{TP}{TP + MP} \text{ and } FPR = \frac{FP}{TP + FP}$$

where TP is the average number of detected regions corresponding to vehicles, MP is the average number of missed

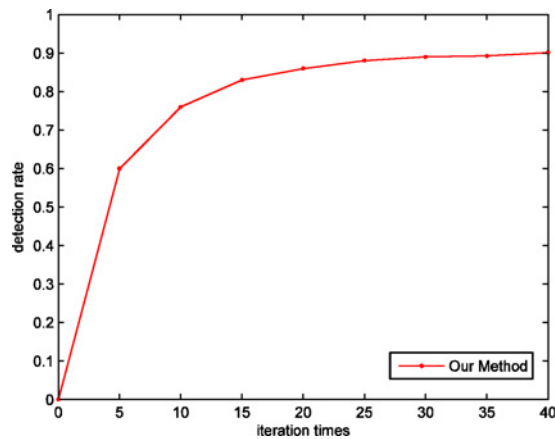


Fig. 9. DR under different number of iterations.

vehicles, and FP denotes the number of detected regions which are actually not vehicles.

C. Parameter Settings

In our experiments, the effects of varying parameter settings in the proposed method are first investigated.

- 1) During bLPS-HOG feature construction, iterations T of each AdaBoost training may influence the performance of the final SVM classifier. The performance of our detector for moving vehicle detection with different iterations T is tested. Fig. 9 shows that the iteration number of 5 is inadequate to take the advantage of the AdaBoost classifier. As the iteration number increases, the DR grows accordingly until the iteration time reaches 25. To achieve satisfactory results, the iteration times should be set to at least 20. Our experiments on several groups of test videos showed that the iteration number $T = 20$ is a good choice for classification.
- 2) F-measure is defined to evaluate the performance of our system under different gradient orientation bins in each block. The F-measure (F) can be interpreted as a weighted average of the precision and recall as follows:

$$F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

where F-measure reaches its best value at 1 and worst at 0. The gradient orientation bin number k in each block also has impact on the system performance. In our experiments, we examined the performance with k from 6 to 21 in the bLPS-HOG feature computing. As shown in Fig. 10, increasing the number of orientation bins improves performance significantly up to about 19 bins, but makes little difference beyond that point. Therefore, $k = 19$ was chosen to use for feature extraction.

D. Experimental Results and Comparison

In order to objectively evaluate the performance of our proposed method, the performance of five other algorithms has been included for comparison. The first algorithm is the simplistic image subtraction method (referred as frame differencing in Table II), which has been widely used in

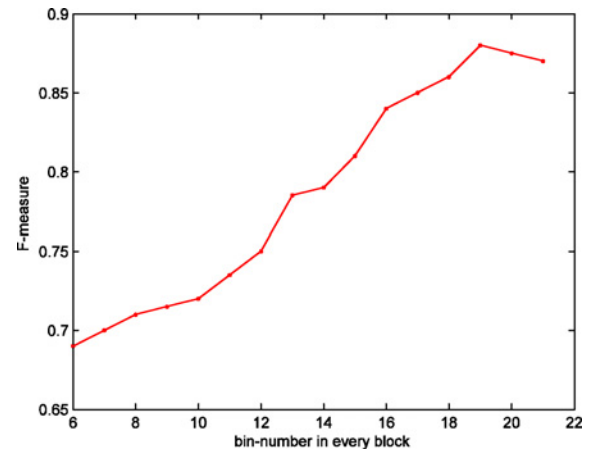


Fig. 10. F-measure value with different number of bins in the boosting LPS-HOG features.

stationary platforms due to its computational efficiency [6], [31]. This method simply subtracts all the pixels in the first frame from the ones in the second frame. The major differences are considered to be caused by moving objects, in our case, moving vehicles. The second method (referred as registration + frame differencing in Table II) performs image subtraction after frame registration [14], which is extended from the simplistic subtraction method but can compensate camera motion. By registering a sequence of frames together, the method can convert a spatio-temporal video segment into a set of spatially corresponded images. Therefore, the motion of airborne platforms can be compensated. The third method (referred as AdaBoost detection + SVM classification in Table II) was proposed in [35], which is performed in two stages. First, AdaBoost classifier using Haar-like features is trained for pixel-wise classification. And then the classified pixels are used for subsequent clustering by SVM based on a set of statistical features. The fourth (referred as bLPS-HOG + voting in Table II) is a voting method based on bLPS-HOG features, which purely counts the number of 1s of the final bLPS-HOG feature. The last one (referred as HOG + SVM in Table II) is similar to our proposed method on vehicle detection, which uses linear SVM classifier but with the original HOG features to perform vehicle classification. All the experiments were carried out using our home-made videos and DARPA VIVID datasets (egtest01.avi and egtest02.avi) on the same hardware platform as described above.

The detection performance of the five methods is presented in Table II. The detection speed was computed as an average of 30 runs. The results show that our method achieved the best performance compared with the other five methods. Although the simplistic image subtraction method shares the advantage of fast detection speed with the proposed method, it produces a large number of false alarms. On the other hand, the registration-based method, which takes too much computational time, cannot satisfy the need of real-time applications. The detection method proposed in [35] yields better results than the first two. The performance is just a little worse than that of our proposed method, since the subsequent clustering after AdaBoost detection is not so good. From Table II, it can be seen that our method outperforms voting in both the DR

TABLE II
COMPARISON OF THE DETECTION PERFORMANCE OF THE SIX METHODS IN OUR HOME-MADE VIDEO AND DARPA VIVID VIDEO TESTS

Video	Metrics	Frame Differencing	Registration + Frame Differencing	AdaBoost Detection + SVM Classification [35]	bLPS-HOG + Voting	HOG + SVM	bLPS-HOG + SVM
Our video	DR (%)	80	86	88	80	88	89
	FPR (%)	45	15	12	23	13	12
	Detection speed (f/s)	33.2	15.6	19.2	25.5	8.15	21.5
DARPA VIVID Video	DR (%)	83	84	86	81	84	88
	FPR (%)	76	23	14	20	14	12
	Detection speed (f/s)	32.5	14.8	18.6	23.8	7.95	20.8

and the FPR. The reason is that the voting method purely depends on the number of 1s in the feature sequence and the permutation does not matter. However, since SVM is a supervised statistical learning method, it is able to exploit the pattern in the feature sequence. In contrast to the SVM classification method using the original HOG features, the significant reduction of feature dimensionality in our proposed method is able to largely speed up the detection process in all video tests. Moreover, our method achieved higher DR with fewer false positives compared to the other five methods mainly due to the following two reasons. First, the used linear SVM classifier, which is trained using the bLPS-HOG feature vectors, has higher classification ability. Second, the STARS measurement is employed to obtain the motion information of the vehicles for motion analysis, which also helps to detect the missed vehicles and remove false positives resulted from the vehicle detection step.

Motion analysis by STARS correlates the same vehicles detected in each video frame, aiming to analyze the motion of the detected vehicles. In order to further evaluate its performance, commonly used methods for tracking moving objects such as Kalman filter, particle filter, and the tracking method proposed in [35] are compared with our proposed STARS metric in terms of tracking rate. These experiments were run on two video segments of our video and *egtest01.avi* in DARPA VIVID datasets. Vehicles in the first frame are initialized by our SVM classifier. As shown in Table III, 356 vehicles in total in various scenarios are tracked using the four methods. The tracking rate is defined as $TR = \frac{NSTV}{Total}$, where *NSTV* refers to the number of successfully tracked vehicles, and *Total* represents the total number of vehicles being tracked. Kalman filter and particle filter used HSV color as the observation model and the constant velocity model as state model. Our proposed STARS metric outperformed all of them, since it has taken spatial and color information into account. As for Kalman and particle filters, the platform in our work is moving. The platform motion can have negative impact on the prediction results of Kalman and particle filters. In addition, it is difficult to choose appropriate state models and measurement models for motion prediction and tracking measurement, respectively, in Kalman and particle filters to get very good performance. The template matching in RGB-color space [35] only takes color information into account, yielding about 94% tracking rate.

To highlight the two major contributions of our proposed method, bLPS-HOG features and STARS-based motion anal-

TABLE III
COMPARISON OF PERFORMANCE OF FOUR METHODS

	Kalman Filter	Particle Filter	Tracking Method [35]	STARS
Total	356	356	356	356
NSTV	325	321	334	343
Tracking rate (%)	91	90	94	96

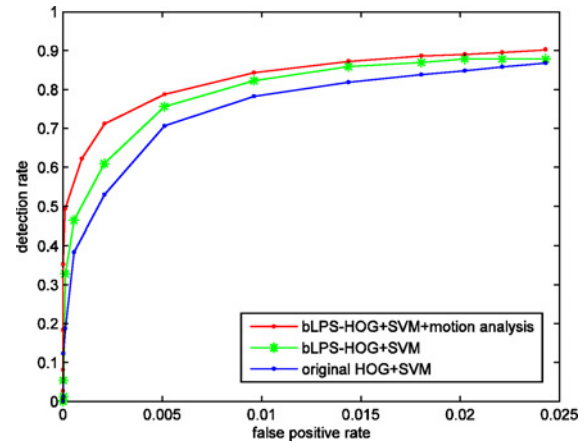


Fig. 11. FPR–DR curve of our method and SVM classification using boosting HOG features and original HOG features.

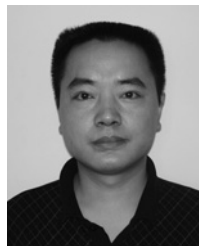
ysis, we further compared the performance of our method and the other two methods (SVM classification based on bLPS-HOG and SVM classification based on original HOG features) using FPR-DR curves on the test videos as shown in Fig. 10. The differences among the three methods are that our method employs bLPS-HOG feature and motion analysis, while the other two methods only uses either the original HOG feature or the bLPS-HOG feature, without motion analysis. As shown in Fig. 11, the SVM classification based on bLPS-HOG features achieved comparable performance with the method based on the original HOG features. On the other hand, the bLPS-HOG feature vector reduces dimensionality significantly, compared with the original HOG features. Thus, the SVM classifier based on bLPS-HOG leads to a much simplified feature vector. Therefore, in our method, the linear SVM classifier is speeded up by using more compact feature vectors. In addition, by motion analysis, missed vehicles can be detected and false positives can be eliminated, leading to a better detection performance.

VI. CONCLUSION

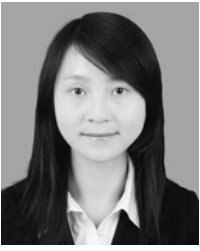
In this paper, we have proposed a method to detect moving vehicles and obtain their trajectories. First, a linear SVM classifier was trained for moving vehicle detection in airborne videos. For the sake of lowering the time cost of the original HOG feature, bLPS-HOG features were proposed and used for SVM training and classification. The obtained potential vehicle regions were further processed to establish the correspondences between moving vehicles in different video frames based on the proposed STARS measure. Our experimental results showed that we achieved high detection performance in addition to the fast detection speed. In our future work, we will design an integrated classification solution for vehicle detection by mining various features to further improve the detection performance of the proposed method.

REFERENCES

- [1] Z. Kim and J. Malik, "Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking," in *Proc. IEEE Int. Conf. Comput. Vision*, vol. 1, Oct. 2003, pp. 524–531.
- [2] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1114–1127, Aug. 2008.
- [3] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Trans. Intell. Transport. Syst.*, vol. 1, no. 2, pp. 119–130, Jun. 2000.
- [4] Y. Wang, "Real-time moving vehicle detection with cast shadow removal in video based on conditional random field," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 437–441, Mar. 2009.
- [5] P. Mirchandani, M. Hickman, A. Angel, and D. Chandnani, "Application of aerial video for traffic flow monitoring and management," in *Earth Observation Magazine*, vol. 12, no. 4, pp. 10–17, Apr. 2003.
- [6] A. Angel and M. Hickman, "Methods of analyzing traffic imagery collected from aerial platforms," *IEEE Trans. Intell. Transport. Syst.*, vol. 4, no. 2, pp. 99–107, Jun. 2003.
- [7] W. Yao, S. Hinz, and U. Stilla, "Automatic vehicle extraction from airborne LiDAR data of urban areas aided by geodesic morphology," *Patt. Recog. Lett.*, vol. 31, no. 10, pp. 1100–1108, 2010.
- [8] A. Yoneyama, C. Yeh, and C. Kuo, "Robust vehicle and traffic information extraction for highway surveillance," *EURASIP J. Appl. Signal Process.*, vol. 2005, no. 14, pp. 2305–2321, 2005.
- [9] K. Khaled, C. Benjamin, P. Claude, and V. Pascal, "A vision algorithm for dynamic detection of moving vehicles with a UAV," in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr. 2005, pp. 1878–1883.
- [10] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.
- [11] E. Line, A. Lars, and K. Hans, "Classification-based vehicle detection in high resolution satellite image," *J. Photogrammetry Remote Sensing*, vol. 64, no. 1, pp. 65–72, Jan. 2009.
- [12] I. Emst, S. Sujew, K. Thiessenhusen, M. Hetscher, S. Rassmann, and M. Ruhe, "LUMOS: Airborne traffic monitoring system," in *Proc. IEEE Intell. Transport. Syst.*, vol. 1, Oct. 2003, pp. 753–759.
- [13] Q. Yu and G. Medioni, "Motion pattern interpretation and detection for tracking moving vehicles in airborne video," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2009, pp. 2671–2678.
- [14] A. C. Shastry and R. A. Schowengerdt, "Airborne video registration and traffic-flow parameter estimation," *IEEE Trans. Intell. Transport. Syst.*, vol. 6, no. 4, pp. 391–405, Dec. 2005.
- [15] I. Cohen and G. Medioni, "Detecting and tracking moving objects in video from an airborne observer," in *Proc. IEEE Image Understand. Workshop*, Nov. 1998, pp. 217–222.
- [16] F. Yamazaki, L. Wen, and T. Vu, "Vehicle extraction and speed detection from digital aerial images," in *Proc. IEEE Int. Conf. Geosci. Remote Sensing Symp.*, vol. 7, Jul. 2008, pp. III-1334–III-1337.
- [17] A. Talukder, S. Goldberg, L. Matthies, and A. Ansar, "Real-time detection of moving objects in a dynamic scene from moving robotic vehicles," in *Proc. IEEE Conf. Intell. Robots Syst.*, vol. 2, Oct. 2003, pp. 1308–1313.
- [18] H. Yalcin, R. Collins, M. Black, and M. Hebert, "A flow-based approach to vehicle detection and background mosaicking in airborne video," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recog.*, Jun. 2005, p. 1202.
- [19] T. Zhao and R. Nevatia, "Car detection in low resolution aerial image," in *Proc. IEEE Int. Conf. Comput. Vision*, vol. 1, Jul. 2001, pp. 710–717.
- [20] B. Lucas and T. Kanade, "Detection and tracking of point features 1981," Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CUM-CS-91-132, Apr. 1991.
- [21] S. M. Khan, H. Cheng, D. Matthies, and H. Sawhney, "3D model based vehicle classification in aerial imagery," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2010, pp. 1681–1687.
- [22] B. Coifman, M. McCord, R. G. Mishalani, M. Iswalt, and Y. Ji, "Roadway traffic monitoring from an unmanned aerial vehicle," *IEE Proc. Intell. Transport Syst.*, vol. 153, no. 1, pp. 11–20, Mar. 2006.
- [23] E. Michaelsen, M. Kirchhof, K. Jager, and U. Stilla, "Classification of local structures in airborne thermal videos for vehicle detection," in *Proc. 3th Int. Symp.: Remote Sens. Data Fusion Urban Areas*, vol. 36, part 8 W27, 2005 [Online]. Available: <http://www.isprs.org/proceedings/XXXVI/8-W27/michaelsen.pdf>
- [24] H. Tao, H. Sawhney, and R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 24, no. 1, pp. 75–89, Jan. 2002.
- [25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recog.*, vol. 1, Jun. 2005, pp. 886–893.
- [26] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vision*, Sep. 1999, pp. 1150–1157.
- [27] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. IEEE Int. Conf. Comput. Vision*, Jan. 1998, pp. 555–562.
- [28] P. Sabzmejdani and G. Mori, "Detecting pedestrians by learning shapelet features," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2007, pp. 1–8.
- [29] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
- [30] Z. R. Wang, Y. L. Jia, H. Huang, and S. M. Tang, "Pedestrian detection using boosted HOG features," in *Proc. IEEE Conf. Intell. Transport. Syst.*, Oct. 2008, pp. 1155–1160.
- [31] A. Angel and M. Hickman, "Methods of traffic data collection using aerial video," in *Proc. IEEE 5th Int. Conf. Intell. Transport. Syst.*, Sep. 2002, pp. 3–6.
- [32] S. Hinz, "Detection of vehicles and vehicle queues in high resolution aerial images," in *Proc. PFG*, Mar. 2004, pp. 201–213.
- [33] S. Ali, V. Reilly, and M. Shah, "Motion and appearance contexts for tracking and reacquiring targets in aerial video," in *Proc. Eur. Conf. Comput. Vision (CVPR)*, 2007, pp. 1–6.
- [34] V. Reilly, H. Idrees, and M. Shah, "Detection and tracking of large number of targets in wide area surveillance," in *Proc. ECCV*, 2010, pp. 186–199.
- [35] D. Rosenbaum, J. Leitloff, F. Kurz, O. Meynberg, and T. Reize, "Real-time image processing for road traffic data extraction from aerial images," in *Proc. ISPRS Commission VII Symp.*, Jun. 2010, pp. 372–388.
- [36] H. X. Jia and Y. J. Zhang, "Fast human detection by boosting histogram of oriented gradients," in *Proc. 4th Int. Conf. Image Graphics*, 2007, pp. 683–688.
- [37] O. Alatas, P. Yan, and M. Shah, "Spatiotemporal regularity flow (SPREF): Its estimation and applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 584–589, May 2007.
- [38] X. Cao, C. Wu, P. Yan, and X. Li, "Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos," to be presented at the IEEE International Conference on Image Processing, Brussels, Belgium, Sep. 2011.



Xianbin Cao (M'08–SM'10) is currently a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China, and is also the Director with the Laboratory of Intelligent Transportation System, Beihang University. His current research interests include intelligent transportation systems, airspace transportation management, and intelligent computation.



Changxia Wu received the B.S. degree from Anhui University, Hefei, China, in 2008, and the M.S. degree from the University of Science and Technology of China, Hefei, in 2011, both in computer science.

Her current research interests include image processing, computer vision, and machine learning.



Jinhe Lan received the B.S. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2009. Currently, he is pursuing the M.S. degree in computer science from the University of Science and Technology of China.

His current research interests include image processing, computer vision, and target recognition and tracking.

Pingkun Yan (S'04–M'06–SM'10) received the B.E. degree in electronics engineering and information science from the University of Science and Technology of China, Hefei, China, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore.

He is a Full Professor with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, Shaanxi, China. His current research interests include computer vision, pattern recognition, machine learning, and their applications in medical imaging.

Xuelong Li (M'02–SM'07) is currently a Full Professor with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, Shaanxi, China.