An Enhanced Viola–Jones Vehicle Detection Method From Unmanned Aerial Vehicles Imagery

Yongzheng Xu, Guizhen Yu, Member, IEEE, Xinkai Wu, Yunpeng Wang, and Yalong Ma

Abstract—This research develops an advanced vehicle detection method, which improves the original Viola–Jones (V-J) object detection scheme for better vehicle detections from low-altitude unmanned aerial vehicle (UAV) imagery. The original V-J method is sensitive to objects’ in-plane rotation, and therefore has difficulties in detecting vehicles with unknown orientations in UAV images. To address this issue, this research proposes a road orientation adjustment method, which rotates each UAV image once so that the roads and on-road vehicles on rotated images will be aligned with the horizontal direction and the V-J vehicle detector. Then, the original V-J can be directly applied to achieve better efficiency and accuracy. The enhanced V-J method is further applied for vehicle tracking. Testing results show that both vehicle detection and tracking methods are competitive compared with other existing methods. Future research will focus on expanding the current methods to detect other transport modes, such as buses, trucks, motorcycles, bicycles, and pedestrians.

Index Terms—Vehicle detection, unmanned aerial vehicle, Viola-Jones, road orientation, vehicle tracking.

I. INTRODUCTION

UNMANNED aerial vehicles (UAVs) hold promise of great value for transportation, as demonstrated by a series of experiments on deploying UAVs for transportation studies [1]–[6]. One important application of UAV technology in transportation is to enhance the traffic and emergency monitoring systems which have been serving as a backbone of Intelligent Transportation System (ITS) infrastructure. Because UAVs are highly portable, UAVs can collect traffic data in the areas where the geographic locations of potential traffic-related problems are only crudely known, or conventional data collection technologies based on point detections cannot be applied to gather the data needed for transportation studies.

For traffic and emergency monitoring, one of the essential but challenging tasks is vehicle detection. Many object detection algorithms have been proposed. Among them, the Viola-Jones (V-J) scheme [7], which combines Adaboost using Haar-like features, has achieved impressive performance and been wildly applied in many fields. However, the original V-J method is sensitive to object orientations; therefore, it can only work when the orientations of vehicles are known. When applying the V-J method for vehicle detection from UAV images, it brings significant challenges because vehicle orientations are generally unknown in UAV images. Although some methods have been proposed to address above issue (e.g., Jones & Viola [8], Cao et al. [9], Leitloff et al. [10], etc.), most of these methods are time-consuming or need extra resources which limit their applications.

This research aims to improve the original V-J scheme so that the enhanced one will be insensitive to on-road vehicles’ in-plane rotation and achieve better accuracy and efficiency. The basic idea is to directly detect the orientation of the road and rotate the road according to the detected orientation only once. The proposed road orientation adjustment method then can be incorporated with the original V-J scheme to achieve better vehicle detection. Since after road orientation adjustment, the majority of vehicle in UAV images will be aligned with the V-J vehicle detectors, the original V-J can be directly applied. More importantly, this method only needs to rotate the image one time, so the new method significantly saves computational time and reduces false detection rates.

The rest of the paper is organized as follows: Section II briefly reviews some work related to vehicle detection methods, followed by the methodological details of the proposed vehicle detection method in Section III. Section IV further expands the proposed vehicle detection method to vehicle tracking. Section V presents a comprehensive evaluation of the proposed road orientation estimation method, the enhanced V-J method, and vehicle tracking method using diverse scenarios. Section VI presents a discussion on some limitations of the proposed vehicle detection method. Finally, Section VII concludes this paper with some remarks.

II. RELATED WORK

A large amount of research has been performed on vehicle detection from UAV images over the years. Many of them apply some traditional methods, such as background subtraction, frame difference, optical flow, etc. For example, Azevedo et al. [11] applied a median-based background subtraction method to fast detect vehicles; Shastry and Schowengerdt [12] applied a frame difference method, combining with the image registration process to
detect moving vehicles; and Yalcin [13] proposed a motion-based optical flow method to detect moving vehicles. However, methods like frame difference, background subtraction and optical flow are sensitive to scene complexity therefore have difficulties in detecting slow-moving or stopped vehicles when traffic is congested. Also, some methods, like optical flow method, are sensitive to background motions.

In recent years, object detection algorithms have become popular for vehicle detections from UAV videos. For example, Cao et al. [14] applied the SVM using HOG features for vehicle detection. To achieve higher efficiency and simpler computation, Cao et al. [9] further proposed a bLPS-HOG feature extraction method with a linear SVM. Leitloff et al. [10] proposed a two-stage method, which first applies the V-J object detection scheme to generate a confidence image and obtained potential regions of vehicles, and then applies a SVM to refine the detection results. Tuermer et al. [15] used Disparity Maps to limit the search space to road regions and then applied a standard detector based on HOG features to detect vehicles. Felzenszwalb [16] developed an objects detection framework by applying discriminatively trained deformable part model (DPM) which is able to represent highly variable object classes with promising object detection results. However, due to the computational complexity, the DPM method is slow in detecting multiple objects therefore cannot satisfy the requirement of real-time applications.

Generally speaking, object detection algorithms are less sensitive to image noise, background motions and scene complexity, therefore are more robust for vehicle detections from UAV videos. However, most of object detection algorithms can only detect vehicles in similar orientations, such as the well-known V-J object detection scheme [7]. Our preliminary experiments [17] show that the V-J framework can only handle approximately ±10 degrees of in-plane rotation. Vehicles with orientations more than ±10 degrees cannot be detected using the original V-J method.

To address the above issue, a simple way is to rotate images multiple times. For example, Cao et al. [9] rotated each video image nine times (each time 20 degree) in order to cover 180 degrees, but repeating detection of the same image significantly increases the detection time and leads to more false detections. Some researchers tried to train multiple detectors for objects of different angles. For example, Jones and Viola [8] built 12 different detectors for faces detection to cover different views. However, training multiple detectors will be a heavy workload. Leitloff et al. [10], on the other hand, used road database to get the road orientation and rotated the image according to the road orientation. Because on-road vehicles run in the same direction with the road, the road and on-road vehicles in the rotated images will be aligned with the original vehicle detector; therefore the original V-J method can be applied. However, the need of additional geometric information limits its applications.

In this paper, instead of using additional road database to get the road orientation, a road orientation adjustment method is developed. Typical road orientation detection methods [18]–[21] are to extract road areas first and then estimate the road orientation by deriving the slope of the center lines of the roads, but these methods are based on the assumption that the color of every pixel in road region is similar. Such assumption could be problematic due to shadows or the change of road surface, leading unreliable detection. This research addresses this issue by proposing a new road orientation detection method which applies the line segment detector (LSD) [22] and relative histogram of line orientations. By incorporating the road orientation adjustment method with the original V-J scheme, an enhanced V-J method is developed. The details of the proposed method are presented in Section III.

### III. Methodology

This proposed vehicle detection method essentially is an improvement of the V-J scheme by incorporating a road orientation adjustment method. The section will start with introducing the theoretical background of the V-J method.

#### A. Viola-Jones Object Detection Scheme

The V-J scheme is based on multiple cascaded Haar-like classifiers [7], [23]. The basic concept is to use a conjunctive set of weak classifiers to form a strong classifier. The core of this scheme is the Haar-like features, which are essentially drawn from the spatial response of Haar basis functions and derivatives to a given type of feature at a given orientation within the image. In practice, Haar-like features are computed as the sum of differences of the pixel intensities between different rectangular regions at a specific location in a detection window (Fig. 1). Rectangular features can be computed very rapidly using an intermediate representation of the image called integral image (also called summed area table) [7]. However, these individual Haar-like features are weak discriminative classifiers, which only give the right answer a little more often than a random decision. To construct a “strong” discriminative classifier, many “weak” classifiers are combined as a conjunctive cascade; and Gentle AdaBoost [24], a machine learning meta-algorithm, is applied to train a cascaded classifier over a set of thousands of positive and negative training images.

The evaluation of the strong classifiers generated by the AdaBoost learning process can be done quickly, but it is not fast enough to process in real-time. For this reason, the strong classifiers are arranged in a cascade in order of complexity. In each cascade, each successive classifier is trained only on those selected samples which pass through the preceding classifiers. If at any stage in the cascade a classifier rejects
the sub-window under inspection, no further processing is performed. The cascade therefore has the form of a degenerate tree. This degenerative decision-tree structure can eliminate negative regions as early as possible during detection to focus attention on promising regions of the image. Therefore this detection strategy dramatically increases the processing speed of the detector, provides an underlying robustness to changes in scale, and maintains achievable real-time performance.

Before applying the V-J method, the first critical step is to build a sample library which provides a training set to include both positive and negative images. As mentioned in [25] and [26], the number of samples plays a key role in training classifier. In our paper, over 16800 positive samples (Fig. 2a) were manually collected from 600 UAV images extracted from 100 minutes of the video data (one image per 10 seconds). From each UAV image, about 10-100 vehicles under different traffic conditions were manually extracted to form 16800 positive training samples. These 16800 samples do not contain duplicated ones and each sample only contains one vehicle. Note that some samples which contain the same vehicle but different backgrounds are treated as different positive samples. All vehicle samples were rotated to align with the horizontal direction. In addition, over 26000 negative samples (Fig. 2b) which do not contain vehicles were also manually collected.

All positive images were further transformed into gray scale images and normalized to a compressed size of 40×20. Each image is used to calculate a complete set of Haar-like features, and in total, 479,430 Haar-like features were extracted from all images. These 479,430 features were further trained into 469 most significant features by applying the Gentle AdaBoost algorithm [24]. Finally, an 18-stage cascaded classifier was formed based on the 469 significant Haar-like features.

**B. Road Orientation Adjustment**

However, when directly applying the V-J method to detect vehicles from UAV images, the detection rate is significantly low. The reason, as mentioned before, is that the original V-J method can only detect vehicles with orientations aligned with the vehicles in training sets. To address this issue, this research proposes a road orientation adjustment method.

Essentially, the proposed road orientation adjustment method is to rotate the image according to the orientation of the road, i.e. the angle between the road and the horizontal of the image. After rotation, the road and on-road vehicles will be aligned with the vehicle detectors. The general procedure includes:

1) Original color images are extracted from aerial videos and transformed into gray scale images (Fig. 3a);

2) The line segment detector (LSD) [22] is applied to detect straight edge segments (Fig. 3b);

3) The orientation of each detected line segment $\theta_i$ is calculated and the relative histogram $H(A)$ of these line orientations is calculated (Fig. 3c). The angle corresponding to the maximum distribution frequency of relative histogram will be the orientation of the road;

4) To minimize in-plane rotation jitters, the final rotation angle $\omega_t$ for frame $t$ is smoothed by the first-order lag filtering algorithm, which considers the rotation angle $\omega_{t-1}$ from the last frame $t-1$.

A rotated image is presented in Fig. 3d. Some technical details are elaborated in the following:

1) **Straight Line Segments Detection.** The proposed method first applies the algorithm of LSD [22] to detect straight line segments. LSD is a linear-time line segment detector which gives subpixel accurate results. This method can work on digital image without parameter tuning, therefore it is more robust and efficient than other line detection algorithms, like Hough transformation, which requires many parameter tunings and is very likely to be affected by other redundant edges. The LSD is open source [27] (available in the Open Source Computer Vision Library, OpenCV, version 3.1). As shown in Fig. 3b, after applying LSD, many line segments were detected. The orientation of each detected line can be estimated by (1):

$$
\varphi_i = \begin{cases} 
\arctan \left( \frac{r_{i2} - r_{i1}}{c_{i2} - c_{i1}} \right), & c_{i1} \neq c_{i2}; \\
90^\circ, & c_{i1} = c_{i2},
\end{cases}
$$

where $\varphi_i$ is the orientation of detected line $i$; $\varphi_i$ is an integer and $\varphi_i \in [0^\circ, 180^\circ]$; $(c_{i1}, r_{i1})$ and $(c_{i2}, r_{i2})$ represent the pixel coordinates of the start and end points (P1 and P2, see Fig. 4) of line $i$ in the image coordinate system.

2) **Road Orientation Estimation by Relative Histogram:** As shown in Fig. 3b, the road is parallel to the majority of the detected line segments. Therefore, to estimate the orientation of the road, essentially, we need to identify the angle of the majority of line segments. Note in practice, we round the angles to $1^\circ$. The relative frequency histogram is applied to identify the angle. The detailed steps are described as the
Fig. 4. Orientation $\phi_i$ of line $i$.

Fig. 5. Suburban road. (a) Color image; (b) Line segments detection using LSD; (c) Relative histogram; (d) Rotated image.

following:

Step 0: Identify the total number of lines;
Step 1: Define 180 class intervals: $\theta_1 = [0^\circ, 1^\circ)$, $\theta_2 = [1^\circ, 2^\circ)$, ..., $\theta_i = [(i - 1)^\circ, i^\circ)$, ..., $\theta_{180} = [179^\circ, 180^\circ)$;
Step 2: Determine the frequency, $h(\theta_i)$, i.e. the number of lines with the angle within the angle interval of class $\theta_i$;
Step 3: Calculate the relative frequency (i.e. proportion) of each class by dividing the class frequency by the total number $n$ in the sample, i.e. $H(\theta_i) = h(\theta_i)/n$;
Step 4: Draw a rectangle for each class with the class interval as the base and the height equal to the relative frequency of the class to form a relative histogram (Fig. 3c);
Step 5: Identify $\theta_k$, which is corresponding to the highest rectangle in relative histogram, and $\Theta = k^\circ$ is considered as the orientation of the road.

3) Rotation Angle Estimation by First-Order Lag Filtering: To minimize the impact of the jitters caused by UAVs, the first-order lag filtering algorithm is further applied to calculate the weighted average of the estimated road orientations of current and previous frames. The final image rotation angle for current frame $j$, is calculated by:

$$\omega_j = (1 - w) \ast \omega_{j-1} + w \ast \Theta,$$

where $\omega_{j-1}$ is the image rotation angle for previous frame $j - 1$, and $w$ is a predetermined weight.

The final step is to rotate the image by $\omega_j$. After rotation, the road will become horizontal. Fig. 5 presents an example for a suburban road.

C. Enhanced Viola-Jones Vehicle Detection Method

With the road orientation adjustment method, a new framework for the enhanced V-J scheme is presented in Fig. 6.

Fig. 6. The frame of the enhanced V-J method.

Fig. 7. Vehicle detections using: (a) the original V-J method; (b) the enhanced V-J method.

A visual comparison of vehicle detections using the original and enhanced V-J methods are shown in Fig. 7. Detected vehicles are marked with red rectangles. As shown in Fig. 7a, because the orientations of some on-road vehicles are about 20 degree from the horizontal, many vehicles could not be detected using the original V-J method. In contrast, in Fig. 7b, by applying the enhanced V-J method, most of vehicles can be detected. Detailed evaluation will be presented in Section V.

IV. APPLICATIONS ON VEHICLE TRACKING

The enhanced V-J method is further applied to improve the accuracy of vehicle tracking. A multiple vehicle tracking framework, which incorporates the proposed enhanced V-J method is proposed. The tracking framework essentially includes two stages: position prediction and color histogram similarities matching. The overall flowchart of the framework is presented in Fig. 8. The framework includes two stages, position prediction and color histogram similarities matching.

Stage 1 Position prediction: The proposed framework first applies a position prediction algorithm, which predicts the
vehicle in time

which essentially defines the cover area of possible positions of tracking vehicles in next frame therefore defined by the V-J scheme. Note the detection window is roughly the size of a vehicle detected in the image, and can be automatically adjusted according to the height of UAV platform and the size of video images during the vehicle detection process. Given that the size of the detection window is \( w \times h \) (\( w \) and \( h \) are the width and height, respectively) and the original relative coordinates of the target vehicle in frame of time \( t \) is \( (x_t, y_t) \), the possible coordinates of the target vehicle in time \( t+1 \) can be defined by the following equation, which essentially defines the cover area of \( R \):

\[
R : \{(x_{t+1}, y_{t+1}) | x_{t+1} \in [x_t \pm \alpha w] \cap [1, \text{ImgW}], y_{t+1} \in [y_t \pm \alpha h] \cap [1, \text{ImgH}],\}
\]

where ImgW and ImgH are the width and height of the image; \( \alpha \) is a scale factor. A typical size for ImgW \( \times \) ImgH is 1920 \( \times \) 1080. (3) basically describes a rectangle area with four corner coordinates of \( (x_t - \alpha w, y_t - \alpha h), (x_t - \alpha w, y_t + \alpha h), (x_t + \alpha w, y_t - \alpha h), \) and \( (x_t + \alpha w, y_t + \alpha h) \), respectively.

**Stage 2 Color histogram similarities matching:** Note with in the area of \( R \), multiple vehicles could be found. So the second stage is to apply a color histogram similarities matching algorithm to find the correct match. The advantage of color histogram similarity method is its invariance to translation and rotation of the image. In addition, color histograms vary slowly with changes of view angle, scale, and occlusion.

For an image \( P \) with \( n \) colors, a histogram of color \( i \), i.e. \( h_i(P) \), the pixel frequency of color \( i \) in the image. \( h_i(P) \) can be calculated by:

\[
h_i(P) = \frac{s_i(P)}{\sum_{j=1}^{n} s_j(P)} \quad (4)
\]

where \( s_i(P) \) is the number of pixel of color \( i \) in image \( P \). With \( h_i(P) \), image \( P \) can be described by a \( n \)-dimensional vector, \( H(P) = [h_1(P), h_2(P), \ldots, h_n(P)] \).

To improve the matching accuracy, four typical color histograms including gray, red (R), green (G), and blue (B) are used to describe an image (see Fig. 9). The color similarity of two images \( P \) and \( P' \), \( S(P, P') \), then can be calculated based on the histogram correlations for all colors. The histogram correlation for color \( i \) between images \( P \) and \( P' \), \( C_i(P, P') \), can be computed by:

\[
C_i(P, P') = \frac{\sum_{j=1}^{N} \left( \left|h_i'(P) - \bar{h}_i(P)\right| \left|h_i'(P') - \bar{h}_i(P')\right| \right)}{\sqrt{\sum_{j=1}^{N} \left|h_i'(P) - \bar{h}_i(P)\right|^2 \left|h_i'(P') - \bar{h}_i(P')\right|^2}}. \quad (5)
\]

where \( N \) is the total number of histogram bins; \( h_i'(P) \) is the frequency of color \( i \) in image \( P \) within histogram bin; \( \bar{h}_i(P) \) is the mean frequency of color \( i \) in image \( P \) of all histogram bins. \( \bar{h}_i(P) = \frac{\sum_{j=1}^{N} h_i'(P)}{N} \). Note \( C_i(P, P') \) should be between 0 and 1. After calculating \( C_i(P, P') \), the color similarity of two images \( P \) and \( P' \), \( S(P, P') \) can be computed:

\[
S(P, P') = \prod_{i \in \{R,G,B,Gray\}} C_i(P, P'). \quad (6)
\]

A match will be identified if \( S(P, P') \) is larger than a pre-determined threshold. Then the system will record and update the position and color histogram of the target vehicle at current time frame and continue to search for matches for the target vehicle in the following frames until the vehicle is out of the camera’s view. Ultimately, the system will derive the trajectory of the target vehicle as shown in Fig. 10. Note multiple vehicles can be tracked simultaneously based on the proposed framework.
It needs to point out it is possible that two vehicles within $R$ in sequential frames have the similar histograms. Under this circumstance, it will affect the tracking accuracy. This is one disadvantage of the proposed tracking method. But based on our observation, the probability of finding two cars which have similar histograms within small searching box $R$ in two sequential frames actually is small. Furthermore, the proposed method uses 4 color histograms to improve the match accurate.

V. EVALUATION

A. UAV Data Collection

The UAV system used in this research is equipped with a quadcopter (model: Phantom 2) airborne platform and a Gopro Hero Black Edition 3 aerial camera (see Fig. 11). A 3-axis gimbal is mounted on the UAV to stabilize the videos and eliminate video jitters caused by UAV therefore greatly reducing the impact from external factors, such as wind. In addition, an On-Screen Display (OSD), an image transmission module and a video monitor are installed in the system for data transmission and airborne flying status monitoring and control.

The performance evaluations of road orientation estimation, vehicle detection, and vehicle tracking were based on low-altitude UAV videos captured from five different scenarios with diverse traffic and weather conditions. These diverse testing scenes are specifically chosen in order to test the robustness of the proposed methods. The details of the UAV video datasets are shown in TABLE I. For each scenario, three 15-min videos were recorded, but only 10-min video in the middle were used due to UAV ascending and descending (Fig. 12). So for each scenario, the dataset includes videos of 30 minutes long. In our experiments, 20 minutes videos of each scenario were chosen for building the sample library and the remainder 10 minutes were used for testing. The resolution and frame rate of the videos are $1920 \times 1080$ and 24 frames per second (fps), respectively. Note, all UAV videos were captured with the UAV hovering over a fixed location. Due to UAV motions, the orientations of the roads and on-road vehicles in the images are unknown and changing frequently.

In addition, for the evaluation of vehicle detection, in order to avoid the situation that the same vehicle in different frames has been detected multiple times, we extract detection images each 20 seconds from 10-minute video. Because the length of the road segment in an image is about 160 meters, most likely a vehicle will pass the road segment in 20 seconds. This could reduce the possibility of one vehicle being detected multiple times. Note when traffic is congested, it is still possible that some slow-moving vehicles will be detected more than once. All experiments are conducted using C++ implementation on a laptop computer (model: ThinkPad T440P) with Intel i5-4300M @ 2.60GHz and 8GB DDR3 memory.

B. Road Orientation Estimation

The accuracy of the proposed road orientation estimation method was evaluated by two typical indicators, mean error ($ME$) and the root mean square ($RMS$), which are defined as:

$$ME = \frac{1}{n} \sum_{i=1}^{n} |\varphi_{est}(i) - \varphi_{GT}(i)|$$
$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\varphi_{est}(i) - \varphi_{GT}(i))^2}$$  (7)

where the error of road orientation estimation is defined as: $\varphi_{est}(i) = \varphi_{est}(i) - \varphi_{GT}(i)$, $\varphi_{est}(i)$ is the estimated road...
orientation of the $i$-th frame, using the proposed road orientation estimation method; $\phi_{GT}(i)$ is the ground truth of road orientation of the $i$-th frame, measured manually. $\phi_{GT}(i)$ is calculated by the Eq. (1) based on the start point and end point of the road centerline. $n$ is the number of UAV images used for measuring the accuracy of the proposed road orientation estimation method.

The evaluation is performed on the video datasets shown in TABLE I. Note for each scene, images were extracted every 30 seconds from the 10-minute video. Therefore, the value of $n$ in Eq. (7) is 20.

TABLE II presents the accuracy of the proposed road orientation estimation method. The results in TABLE II demonstrate high accuracy of the proposed method. The maximum $ME$ and $RMS$ among all the five scenes are 1.63° and 2.15°, respectively. The mean error $ME$ was mainly caused by two reasons:

1) Rounding error. When the orientation $\phi_i$ of one detected line $i$ is rounded to an integer number by the Eq. (1), error due to rounding will be produced;

2) Image distortion. Due to the camera lens distortion, the road boundaries, centerlines and outlines in airborne images are not straight (slightly curved). For this reason, some detected line segments which are parts of the road boundaries, centerlines and outlines will be in different directions (note that those detected line segments will be strictly paralleled to each other if image distortion is rectified).

However, the maximum $\phi_{err}$ (5.52°) of road orientation estimation (in TABLE II) has little influence on vehicle detection, as the V-J method can handle approximately ±10 degrees of in-plane rotation according to our preliminary experiments [17].

### TABLE II

<table>
<thead>
<tr>
<th>Metrics</th>
<th>freeway</th>
<th>urban road 1</th>
<th>urban road 2</th>
<th>urban road 3</th>
<th>urban road 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ME$ in °</td>
<td>0.69</td>
<td>1.63</td>
<td>0.34</td>
<td>1.57</td>
<td>0.34</td>
</tr>
<tr>
<td>$RMS$ in °</td>
<td>1.17</td>
<td>2.15</td>
<td>0.42</td>
<td>2.00</td>
<td>0.46</td>
</tr>
<tr>
<td>maximum $\phi_{err}$ in °</td>
<td>3.41</td>
<td>5.52</td>
<td>0.80</td>
<td>4.88</td>
<td>1.10</td>
</tr>
</tbody>
</table>

C. Vehicle Detection

1) Evaluation Performances: Four performances (or indicators) are chosen to evaluate the detection accuracy including: Detection speed in terms of frames per second (fps), Correctness (Cor.), Completeness (Com.), and Quality (Qua.), as defined in (8):

\[
\text{Cor.} = \frac{TP}{TP + FP}, \quad \text{Com.} = \frac{TP}{TP + FN}
\]

\[
\text{Qua} = \frac{TP}{TP + FP + FN}
\]

where $TP$ is the number of “true” detected vehicles; $FP$ is the number of “false” detected objects which are non-vehicle objects; and $FN$ is the number of vehicles missed. In particular, Quality is considered as the strictest criterion, which contains both possible detection errors (false positives and false negatives). Note that a successful “detection” is defined as a correct detection of a vehicle in one frame.

2) Results and Comparison: The enhanced V-J, which incorporates the road orientation adjustment method, is insensitive to road orientation changes, so conceptually it should be more effective for on-road vehicle detections. To verify this point, a comprehensive comparison, which compares the proposed method with the following six other methods, is conducted here:

1) Original V-J method,
2) Rotate each image every 20° from 0° to 180° and detect 9 times using the original V-J method (referred as V-J + rotate 9 times in TABLE III) [9],
3) Linear SVM classifier with HOG feature (referred as HOG + SVM in TABLE III) [28],
4) ViBe, a universal background subtraction algorithm [29],
5) Frame difference [12], and
6) Optical flow [30].

Because ViBe [29], frame difference [12], and optical flow [30] are sensitive to background motions, the image registration process [12] is performed first to compensate UAV motions. This pro-processing process converts the spatio-temporal video into temporal information, thereby can correct UAV motion and attitude errors. The image registration processing time is included in the detection time for these three methods.

As mentioned above, for each scenario, a 10-min video with the resolution of 1920×1080 was used for testing. The detection speed for each method was computed as an average of each 10-min video. Note vehicle detection was performed on the entire image (1920×1080). As mentioned before, detection images were extracted every 20 seconds, so for each scenario, totally of 30 detected images were extracted and used for computing Correctness, Completeness, and Quality.

The testing results of seven methods are presented in TABLE III. The average metrics listed in the bottom of TABLE III show that our method achieved the best Quality (82.17%) compared with the other six methods. ViBe [29] and frame difference [12] achieved fast detection speed but with low Quality (54.24% & 49.03%). This is because some non-vehicle objects (such as tricycles and moving pedestrians) lead to many false positives. Besides, slow-moving or stopped vehicles and some black vehicles which have similar colors with the road surface cannot be detected. The Quality (57.91%) of optical flow [30] is slightly better than frame difference [12] and ViBe [29] but with slow speed (0.85 fps) due to its huge computation.

The performance of the HOG + SVM [28] is with a Quality of 64.32% and a detection speed of 1.07 fps. The Completeness of the HOG + SVM [28] is low (73.01%), due to that when the road orientations are unknown and varying, some vehicles that are not in the horizontal directions cannot be detected, leading many false negatives.

The performance of the original V-J vehicle detector with a Quality of 66.79% and a detection speed of 1.14 fps is better than HOG + SVM and many other methods. This is the reason why we choose the V-J scheme as the basis for our method.
TABLE III

<table>
<thead>
<tr>
<th>Scene</th>
<th>Metrics</th>
<th>V-J</th>
<th>V-J + rotate 9 times</th>
<th>HOG + SVM</th>
<th>Registration+ Vibe</th>
<th>Registration+ Frame difference</th>
<th>Registration+ Optical flow</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td>freeway</td>
<td>Correctness (%)</td>
<td>85.71%</td>
<td>87.50%</td>
<td>91.30%</td>
<td>65.00%</td>
<td>61.11%</td>
<td>80.00%</td>
<td>88.57%</td>
</tr>
<tr>
<td></td>
<td>Completeness (%)</td>
<td>75.00%</td>
<td>92.11%</td>
<td>55.26%</td>
<td>86.67%</td>
<td>78.57%</td>
<td>84.21%</td>
<td>93.94%</td>
</tr>
<tr>
<td></td>
<td>Quality (%)</td>
<td>66.67%</td>
<td>81.40%</td>
<td>52.50%</td>
<td>59.09%</td>
<td>52.38%</td>
<td>69.57%</td>
<td>83.78%</td>
</tr>
<tr>
<td></td>
<td>Detection Speed (fps)</td>
<td>1.16</td>
<td>0.064</td>
<td>1.17</td>
<td>5.59</td>
<td>9.96</td>
<td>0.85</td>
<td>1.04</td>
</tr>
<tr>
<td>urban road 1</td>
<td>Correctness (%)</td>
<td>89.47%</td>
<td>73.53%</td>
<td>82.05%</td>
<td>49.06%</td>
<td>77.78%</td>
<td>81.82%</td>
<td>88.00%</td>
</tr>
<tr>
<td></td>
<td>Completeness (%)</td>
<td>70.83%</td>
<td>92.59%</td>
<td>69.57%</td>
<td>66.67%</td>
<td>48.84%</td>
<td>78.26%</td>
<td>91.67%</td>
</tr>
<tr>
<td></td>
<td>Quality (%)</td>
<td>65.38%</td>
<td>69.44%</td>
<td>60.38%</td>
<td>39.39%</td>
<td>42.86%</td>
<td>66.67%</td>
<td>81.48%</td>
</tr>
<tr>
<td></td>
<td>Detection Speed (fps)</td>
<td>2.08</td>
<td>0.16</td>
<td>1.15</td>
<td>6.64</td>
<td>10.40</td>
<td>0.82</td>
<td>1.59</td>
</tr>
<tr>
<td>urban road 2</td>
<td>Correctness (%)</td>
<td>85.06%</td>
<td>92.16%</td>
<td>98.67%</td>
<td>77.87%</td>
<td>75.26%</td>
<td>84.00%</td>
<td>95.65%</td>
</tr>
<tr>
<td></td>
<td>Completeness (%)</td>
<td>89.16%</td>
<td>94.95%</td>
<td>77.08%</td>
<td>86.36%</td>
<td>68.87%</td>
<td>65.63%</td>
<td>94.62%</td>
</tr>
<tr>
<td></td>
<td>Quality (%)</td>
<td>77.08%</td>
<td>87.85%</td>
<td>76.29%</td>
<td>69.34%</td>
<td>56.15%</td>
<td>58.33%</td>
<td>90.72%</td>
</tr>
<tr>
<td></td>
<td>Detection Speed (fps)</td>
<td>0.50</td>
<td>0.049</td>
<td>1.02</td>
<td>6.17</td>
<td>10.01</td>
<td>0.83</td>
<td>0.48</td>
</tr>
<tr>
<td>urban road 3</td>
<td>Correctness (%)</td>
<td>64.29%</td>
<td>40.54%</td>
<td>66.67%</td>
<td>51.11%</td>
<td>53.49%</td>
<td>64.71%</td>
<td>77.78%</td>
</tr>
<tr>
<td></td>
<td>Completeness (%)</td>
<td>69.23%</td>
<td>82.24%</td>
<td>84.21%</td>
<td>79.31%</td>
<td>85.19%</td>
<td>78.57%</td>
<td>93.33%</td>
</tr>
<tr>
<td></td>
<td>Quality (%)</td>
<td>50.00%</td>
<td>38.46%</td>
<td>59.26%</td>
<td>45.10%</td>
<td>48.94%</td>
<td>55.00%</td>
<td>73.68%</td>
</tr>
<tr>
<td></td>
<td>Detection Speed (fps)</td>
<td>1.37</td>
<td>0.063</td>
<td>1.16</td>
<td>6.22</td>
<td>10.26</td>
<td>0.86</td>
<td>1.06</td>
</tr>
<tr>
<td>urban road 4</td>
<td>Correctness (%)</td>
<td>91.59%</td>
<td>86.76%</td>
<td>99.99%</td>
<td>67.89%</td>
<td>77.53%</td>
<td>86.67%</td>
<td>90.32%</td>
</tr>
<tr>
<td></td>
<td>Completeness (%)</td>
<td>80.33%</td>
<td>93.65%</td>
<td>78.91%</td>
<td>80.43%</td>
<td>51.49%</td>
<td>42.62%</td>
<td>88.89%</td>
</tr>
<tr>
<td></td>
<td>Quality (%)</td>
<td>74.81%</td>
<td>81.94%</td>
<td>73.19%</td>
<td>58.27%</td>
<td>44.81%</td>
<td>40.00%</td>
<td>81.16%</td>
</tr>
<tr>
<td></td>
<td>Detection Speed (fps)</td>
<td>0.60</td>
<td>0.056</td>
<td>0.85</td>
<td>5.27</td>
<td>9.64</td>
<td>0.89</td>
<td>0.52</td>
</tr>
<tr>
<td>average</td>
<td>Correctness (%)</td>
<td>83.22%</td>
<td>76.10%</td>
<td>85.94%</td>
<td>62.19%</td>
<td>69.03%</td>
<td>79.44%</td>
<td>88.06%</td>
</tr>
<tr>
<td></td>
<td>Completeness (%)</td>
<td>76.91%</td>
<td>92.31%</td>
<td>73.01%</td>
<td>79.89%</td>
<td>66.59%</td>
<td>69.86%</td>
<td>92.49%</td>
</tr>
<tr>
<td></td>
<td>Quality (%)</td>
<td>66.79%</td>
<td>71.82%</td>
<td>64.32%</td>
<td>54.24%</td>
<td>49.03%</td>
<td>57.91%</td>
<td>82.17%</td>
</tr>
<tr>
<td></td>
<td>Detection Speed (fps)</td>
<td>1.14</td>
<td>0.079</td>
<td>1.07</td>
<td>5.98</td>
<td>10.05</td>
<td>0.85</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Fig. 13. Roads orientation detection for an interchange: (a) Color image, (b) Line segments detection using LSD; (c) Relative frequency histogram.

Fig. 14. Roads orientation detection for an intersection: (a) Color image, (b) Line segments detection using LSD; (c) Relative frequency histogram.

However, because the V-J method is also sensitive to vehicles’ in-plane rotation, the Completeness is only 76.91%.

For the V-J + rotate 9 times, the Completeness (92.31%) is significantly higher than that of the original V-J (76.91%) and HOG + SVM methods (73.01%). This is because when images were rotated every 20° from 0° to 180° and detected 9 times, theoretically, vehicles of different orientations can be fully covered. However, repeating detections of the same image lead to more false positives and greatly increase the detection time. The Correctness of 76.10% is lower than that of the original V-J (83.22%). The detection speed of 0.079 fps is 14.4 times slower than that of the original V-J (1.14 fps).

The enhanced V-J yields the best Completeness (92.49%) of all the seven methods. It is better than the method of V-J + rotate 9 times (Completeness, 92.31%), because by incorporating the road orientation adjustment method, on-road vehicles of unknown orientations will be aligned with the horizontal direction which can be detected using the original V-J vehicle detector. The enhanced V-J is better than the V-J + rotate 9 times in Completeness, because those rotated vehicles...
TABLE IV
VEHICLE TRACKING RESULTS

<table>
<thead>
<tr>
<th>Scene</th>
<th>Metrics</th>
<th>Particle Filter</th>
<th>Kalman Filter</th>
<th>Template Matching</th>
<th>KLT tracker</th>
<th>Software [34]</th>
<th>KCF</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correctness (%)</td>
<td>89.47%</td>
<td>92.11%</td>
<td>90.00%</td>
<td>91.18%</td>
<td>87.80%</td>
<td>92.31%</td>
<td>94.29%</td>
</tr>
<tr>
<td></td>
<td>Completeness (%)</td>
<td>91.89%</td>
<td>94.59%</td>
<td>94.74%</td>
<td>83.78%</td>
<td>97.30%</td>
<td>97.30%</td>
<td>89.19%</td>
</tr>
<tr>
<td></td>
<td>Quality (%)</td>
<td>82.93%</td>
<td>87.50%</td>
<td>85.71%</td>
<td>77.50%</td>
<td>85.71%</td>
<td>90.00%</td>
<td>84.62%</td>
</tr>
<tr>
<td>Tracking Speed (fps)</td>
<td>4.35</td>
<td>4.87</td>
<td>4.63</td>
<td>2.86</td>
<td>0.88</td>
<td>3.97</td>
<td>5.26</td>
<td></td>
</tr>
</tbody>
</table>

**urban road 1**

| Correctness (%) | 83.56% | 88.24% | 94.03% | 89.83% | 88.46% | 89.39% | 95.31% |
| Completeness (%)  | 95.31% | 93.75% | 98.44% | 82.81% | 85.19% | 92.19% | 95.31% |
| Quality (%)      | 80.26% | 83.33% | 92.65% | 75.71% | 76.67% | 83.10% | 91.04% |
| Tracking Speed (fps) | 4.76 | 5.65 | 4.39 | 1.94 | 0.57 | 3.76 | 5.05 |

**urban road 2**

| Correctness (%) | 93.75% | 91.36% | 97.56% | 93.83% | 91.86% | 95.24% | 98.65% |
| Completeness (%)  | 91.46% | 90.24% | 97.56% | 92.68% | 96.34% | 97.56% | 89.02% |
| Quality (%)      | 86.21% | 83.15% | 95.24% | 87.36% | 88.76% | 93.02% | 87.95% |
| Tracking Speed (fps) | 3.42 | 3.82 | 2.79 | 1.27 | 0.27 | 2.17 | 4.12 |

**urban road 3**

| Correctness (%) | 96.48% | 94.44% | 95.45% | 97.76% | 97.86% | 98.84% | 98.53% |
| Completeness (%)  | 99.28% | 98.55% | 91.30% | 94.93% | 99.28% | 98.55% | 97.10% |
| Quality (%)      | 95.80% | 93.15% | 87.50% | 92.91% | 97.16% | 96.45% | 95.71% |
| Tracking Speed (fps) | 6.45 | 7.48 | 5.29 | 2.71 | 0.56 | 3.31 | 6.06 |

**average**

| Correctness (%) | 86.59% | 87.88% | 90.28% | 90.00% | 90.73% | 87.95% | 98.46% |
| Completeness (%)  | 94.04% | 96.03% | 86.09% | 89.40% | 95.80% | 96.69% | 84.77% |
| Quality (%)      | 82.08% | 84.80% | 78.79% | 81.33% | 87.26% | 85.38% | 83.66% |
| Tracking Speed (fps) | 3.76 | 5.87 | 2.93 | 1.24 | 0.24 | 1.70 | 6.03 |

in the V-J + rotate 9 times are in fact not exactly aligned with the horizontal direction, therefore cannot adapt the original V-J method well.

Furthermore, because the road orientation adjustment is fast (60 ms per frame) and each image only needs to be rotated once, the enhanced V-J method is efficient (0.94 f/s). Overall, the enhanced V-J method achieves good vehicle detection performance and maintains the advantage of fast detection of the original V-J scheme. Besides, the proposed method can also be performed on videos captured from moving UAV platforms (for example, UAV flying along the road) without the need of image registration [12], [29], [30] or additional road database [10], [15], thus it has great potentials of wild field applications. The enhanced method achieves the best Quality (82.17%) over other methods. We have to point out that the enhanced method is slower than the original V-J vehicle detector, HOG + SVM, Vibe [29] and frame difference [12], but detection speed can be accelerated by using better graphics processing unit (GPU), parallel computation, or running on high performance computers.

### D. Vehicle Tracking

1) **Evaluation Performances**: The performance of tracking is also evaluated by four indicators: Tracking speed, Correctness (Cor.), Completeness (Com.), and Quality (Qua.), as explained in Eq. (8). Note in the equation, \(TP\) is the number of true tracked vehicles; \(FP\) is the number of false tracked objects which are non-vehicle objects; and \(FN\) is the number of vehicles which are failed to be tracked. Note a successful tracking is defined as tracking a vehicle through the whole image without tracking disruption.

2) **Results and Comparison**: To conduct a comprehensive evaluation, the proposed tracking framework is compared to six other tracking methods including: 1) particle filter [31], 2) Kalman filter, 3) template matching based on correlation coefficient, 4) Kanade – Lucas - Tomasi (KLT) feature tracker [32], 5) the state-of-the-art tracking algorithm Kernelized Correlation Filters (KCF) [33], and 6) a software package [34], which tracks moving objects based on a group of KLT feature trackers.

The evaluation is performed on the same five datasets which were used for vehicle detection evaluation (V-B). Note that in our experiments, the tracking tests are only performed on the road regions, so the size of the tracking area is set 1200×400. The tracking speed for each method was computed as an average of each 10-min video. Note the tracking time for the testing includes the process time for image registration.

The testing results of seven methods are shown in TABLE IV. The average metrics are listed in the bottom of TABLE IV. The proposed framework achieved an average Quality of 88.60% with an average tracking speed of 5.30 f/s. Compared with other methods, the Quality (88.60%) of our method is slightly lower than KCF (Quality, 89.59%), but the tracking speed of our method (5.30 f/s) is much faster than KCF (2.98 f/s). The Kalman filter has faster tracking speed (5.54 f/s), but with a lower Quality (86.39%). In addition, the Kalman filter is sensitive to vehicle speed variations. Also, during the application, it is difficult to choose appropriate state models for motion prediction. The color-based particle filter [31] which uses HSV color distributions as target models can be easily influenced by neighboring vehicles with similar color and appearances. The template matching method based on correlation coefficient achieves lower performance than our
method in terms of Quality (87.98%) and speed (3.94 f/s). The KLT tracker achieved good performance (Quality of 82.96%, and speed of 2.00 f/s) by tracking one salient feature such as corner points on vehicles; but once the feature tracking is disrupted, the vehicle tracking will fail. The software package [34] indeed is an improved method of KLT tracker. Instead of tracking one feature on the vehicle, [34] tracks a group features. The advantage of [34] is that tracking disruption of a single feature will not influence the whole tracking process. Therefore, [34] achieved a much higher Completeness (94.78%) than the KLT tracker (88.72%) and many other methods except KCF (96.46%). However, due to the great computation time, [34] has the slowest speed (0.50 f/s) among all seven methods. Overall, the proposed framework demonstrates competitive performance compared to other six methods.

VI. DISCUSSION

A. Road Orientation Adjustment for Roadways With More Than One Orientation

A highlight of the research is the road orientation adjustment method, which is general and can also be applied for roadways with more than one orientation. Here we present some testing results for roads with two orientations. As shown in Fig. 13 and Fig. 14, the proposed road orientation adjustment method was applied to: 1) an interchange with one freeway crossing over an arterial, and 2) a regular 4-leg intersection. As shown in the figures, two peaks were found in the relative frequency histograms. These two peaks essentially indicate the orientations of the roads. For the case of freeway interchange (Fig. 13), the orientation for the arterial is around 90° and the orientation for the freeway is around 0°; and for the case of 4-leg intersection (Fig. 14), the orientations for the two approaches are around 0° and 92°, respectively. Note conceptually the method can be applied to detect many orientations; but when the road orientations are more than two, the peaks in the relative histogram could be difficult to identify.

B. Vehicle Detection Shortages – Influence of Illumination & Turning Vehicles

As argued by some research [35], the V-J method is sensitive to lighting conditions. So the captured images should be pre-processed to overcome illumination variations. Our method cannot conquer this drawback since it follows the original V-J scheme. But it would be interesting to see the performance of our method under lighting conditions without pre-processing. We present a testing result using the 10-minute video captured from scene “urban road 5” in which vehicles were traveling from an illumination (or shadowed) area to a shadowed (or illumination) area. The data description of scene “urban road 5” is described as: Non-congested urban road with shadows; data recorded at: North Fourth Ring Road, Beijing, China; from 16:00 to 16:15, November 6, 2014; flight altitude 150 meters. Note for this scenario, a 15-min video is recorded. Due to UAV ascending and descending, only 10-min video in the middle was used for testing.

The testing results are highlighted in TABLE V. The results show that, the proposed method achieved a lower Completeness (81.36%) than the average level (92.49%), due to that many black vehicles cannot be detected, as shown in Fig. 15 (detected vehicles are marked with red rectangles). The results confirm that lighting has a strong impact on the accuracy of vehicle detection, but the Completeness of 81.36% is still acceptable. One potential way to address this issue is to collect more training samples from illumination changing scenes to train different detectors.

Another drawback of the proposed vehicle detection method is its incapability of detecting turning vehicles. The roadway orientation adjustment can only rotate the image according to the orientation of the road. For turning vehicles, their orientations are changing during turning process and not aligned with the V-J detector. This creates difficulties of vehicle detection. The original V-J method also has this problem. Combining a sophisticated vehicle tracking method might be a potential solution to address this problem.

VII. CONCLUDING REMARKS

In this paper, an enhanced V-J vehicle detection method using low-altitude UAV images is proposed. The original V-J scheme cannot detect vehicles with unknown orientations. To address this issue, this research first developed a road orientation adjustment method, which rotates each UAV image so roads and on-road vehicles in images are aligned with the horizontal direction and the V-J vehicle detector. Because the road orientation adjustment is fast and each image only needs to be rotated once, the enhanced V-J method achieves promising performance on both static and moving UAV platforms.

However, the enhanced V-J method still has difficulties to address the illumination and turning problems. The future research will be focusing on solving these problems.
Furthermore, the current research only focuses on detecting regular passenger vehicles. Future research will be focusing on expanding the current method for detecting other transportation modes such as buses, trucks, motors, bicycles, and pedestrians. Also, robust vehicle detection in various open environments will be one of our future research directions.

ACKNOWLEDGMENT

The authors would like to thank the insightful and constructive comments from the anonymous reviewers.

REFERENCES


Yongzheng Xu received the B.S. degree in automobile engineering and the M.S. degree in transportation engineering from Shandong University of Technology in 2006 and 2012, respectively. He is currently working toward the Ph.D. degree in transportation with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His research interests include UAV and traffic data collection.
Guizhen Yu (M’09) received the Ph.D. degree from Jilin University, China, in 2003. He is an Associate Professor with the School of Transportation Science and Engineering, Beihang University, China. His research interests include intelligent vehicle and urban traffic operations. He is a member of the IEEE ITS Society.

Xinkai Wu received the Ph.D. degree from University of Minnesota Twin Cities in 2010. He is currently a Professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His research interests include urban traffic operations, electric vehicles, and ITS.

Yunpeng Wang received the Ph.D. degree from Jilin University, China, in 1997. He is a Professor with the School of Transportation Science and Engineering, Beihang University, China. His research interests include cooperative vehicle infrastructure systems and intelligent transportation control.

Yalong Ma received the B.S. degree in traffic engineering from Hefei University of Technology, Hefei, China, in 2011. He is currently working toward the master’s degree with an emphasis on transportation at the School of Transportation Science and Engineering, Beihang University, Beijing, China. His research interests include intelligent transportation systems and computer vision.