Self-Learning Vehicle Detection and Tracking from UAVs

Xiyan Chen and Qinggang Meng
Loughborough University Department of Computer Science, Loughborough, UK
Email: X.Chen-09@student.lboro.ac.uk, Q.Meng@lboro.ac.uk

***Abstract*—Vehicle detection and tracking with unmanned aerial vehicles (UAVs) find increasingly widespread applications in both military and civilian domains. In this paper, a method for vision-based multiple vehicle detection by using the proposed self –learning tracking and detection (SLTD) method has been proposed. The method used Features from Accelerated Segment Test (FAST) and Histograms of Oriented Gradients (HoG) for vehicle detection. Based on the detection results, a Forward and Backward Tracing (FBT) mechanism has been employed in the new self-learning tracking algorithm based on Scalar Invariant Feature Transform (SIFT) feature. The main aim of this research is to improve the accuracy of the detection and tracking system, where the detector relies on the features of a pre-trained model with no connection with the current detection or tracking. The main contribution of this paper is that the proposed system can detect and track multiple vehicles with a self-learning process leading to increase the tracking and detection accuracy. UAV videos captured in different situations have been used to evaluate the proposed algorithm. The results demonstrated that the accuracy can be improved by using the proposed method.**

***Index Terms*—Unmanned Aerial Vehicle (UAV), vehicle detection, self-learning-tracking, Forward and Backward Tracking (FBT)**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have become a new area in aviation industry and are deployed in both military and civilian applications. UAVs have advantages of "zero" casualties, high mobility, fast deployment and wide surveillance scope, as well as being capable of deployment in extreme environments and situations. In particular, vehicle detection from aerial images or videos has become a key research topic in many areas such as automatic traffic monitoring, commercial aerial surveillance and security related tasks, etc. Normally UAVs are remotely controlled by an operator with a terminal device that receives the aerial images taken from the UAVs. In this circumstance the operator needs to monitor and examine the images/videos by themselves and then make the decision to control the UAVs.

In recent years there has been increased research in autonomous vehicle detection and tracking by UAVs. One of the main challenges of the detection and tracking is the target objects might change their appearance or reappear during the tracking process which might cause errors. The detection and tracking process also needs to handle the following various problems. First of all, the tracker and detector should be scale invariant to the targets which can reduce the errors caused by the UAVs changing their altitude during tracking. Secondly, the UAV's flight directions change rapidly and unpredictably, leading the changing directions of the target's movement. Thus, rotationally invariant features are needed. Furthermore, the illumination to the target may vary depending on the UAV's flight directions and the shooting angles to the target. Also the images captured by the UAV camera may get blurred and warped so they need to get transformed to reduce these effects. Therefore the transformation invariant is needed. Furthermore, the background confusions and targets occlusions may exist. Finally, the most important issue is the detection and tracking process have to be real-time.

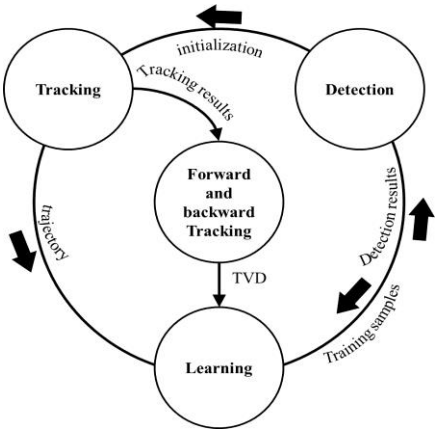


Figure 1. The diagram of the proposed approach

In this paper, we propose a method of self-learning UAV vehicle detection and tracking as shown in Fig. 1. From the input video, vehicles are detected by using the features extracted from Histogram of Oriented Gradients (HoG) [1] and Features from Accelerated Segment Test (FAST) [2] with Support Vector Machine (SVM) classifier [3]. It is assumed that the vehicle has higher density of corners than other objects in the environment so finding the distribution of corners should be the very first thing to narrow the area for further HoG processing. FAST corner detection method can quickly and accurately detect relevant corner points. In this system, the testing

Manuscript received September 15, 2015; revised December 17, 2015.

doi: 10.18178/ijmerr.5.2.149-155

results of the Region of Interest (RoI) by using FAST achieved accuracy of 98.3%. The proposed method applied the HoG algorithm to detect the edge pattern in a rectangular shape which is the most obvious feature for vehicles. However, the HoG is not rotationally invariant so the proposed approach tackles this by using four separate orientation directions (0°, 45°, 90°, and 135°) into the original HoG algorithm. One of the key elements in the system is tracking as shown in Fig. 1. The original Tracking Learning Detection (TLD) in [4] can only track one object, the proposed approach in this paper extended it to track multiple targets in real-time. It is assumed that both detection and tracking process could make errors during the process so the detection and tracking processes should monitor each other during the process and the system keeps updating the detection and tracking results.

Another difference from the original TLD algorithm is that a Forward and Backward Tracking (FBT) model has been proposed between the detection and tracking process to check if there are any errors in the results of each of detection and tracking processes. Two inspectors (positive and negative) have been developed for the error estimations which will be discussed in Sector IV-B. Furthermore, the FBT will also update the classification samples based on the tracking results for future detection use. There are two assumptions in the FBT. Firstly the FBT checks the current tracking results with the results from previous and following frames, also with the results in the current detection process. In an image frames sequence, the matching score between the same trackers should be very high when the tracker is tracking the same target. The FBT has a Tracked Vehicle Database (TVD) which stores the SIFT information about already tracked vehicles. If the FBT results indicated that all matching results between the trackers in the frame sequences are from the same vehicle which means the tracker is tracking the same object, the features of this object will then be checked with the results in the detection process. If they are very similar, this means the tracker is tracking the same object. On the other hand, if the vehicles being tracked are different from the detected vehicle, this will be considered as false negative error for the tracking which means a new vehicle has been detected and tracked, so a new tracker will be created for tracking the new detected vehicle. All these results in the FBT will be considered as positive samples used later in classification. Secondly, the other assumption of the FBT is if the tracker cannot match with both previous and following frames this tracking result will be considered as the false negative sample for later detection training. This approach applied Scale-Invariant Feature Transform (SIFT) matching method in the tracking process because SIFT has a considerable high matching performance with acceptable processing resources requirement. In the TVD, each vehicle has its own SIFT points' descriptors which will be used in the matching process.

The rest of this paper is organized as follows: Section II is the literature reviews of the work related to vehicle tracking and detection. Section III introduces the vehicle detection method; Section IV presents vehicle tracking methods; Section V presents the experiments and the results; and Section VI concludes the paper.

II. Related Work

Vehicle detection methods can be broadly divided into two groups which are appearance-based and motion-based methods. Appearance-based methods recognize vehicles directly from a single image and the motion-based methods require a set of sequenced images or frames of a video in order to recognize vehicles. Generally speaking, the appearance-based methods are more common used in the literature because the motion-based methods are only suitable in moving object detection with a stable camera. In this situations, where videos are captured from a flying UAV which means the stationary vehicles cannot be recognized apart from the background by the motion-based method, so appearance-based methods were used in the proposed detection process. The appearance-based target detection are typically based on two methods, local object features method [5] and the sliding window method [6]. The local feature method always has three main principles: feature abstract, feature classification and model fitting. The main advantages of this method is to perceive the object feature in advance, however it is also a drawback which means it can only detect the already-known feature classes. The sliding window method works as scanning the whole input image by a pre-set window in a certain size and each time the method will decide whether the current sub-image contains the target object or not. This method requires large computational calculations which might be a weakness for the real-time process.

HoG based approach is a commonly used in the appearance-feature-based vehicle detection. HoG features are extracted by evaluating edge operators over the whole image and discretizing and binning the orientations of the edge intensities into histogram descriptors that are used for creating classification models. Shi *et al.* [7] developed a context-driven framework to improve the detection of moving vehicles. Their approach comprises three stages: motion detection, vehicle detection and an online road network estimation filter. The HoG features are used in cascade SVM classifiers to detect vehicles. They evaluated their system and obtained a positive target and negative background classification rate of 84.3% and 79.7% respectively for their detections. Gleason *et al.* [8] compared the performance of HoG feature and Histogram of Gabor Coefficients (HGC) features used as the descriptors of vehicles, it obtaining an average detection rate of 80%. According to the detection rate figures the HoG has obtained better performance. They also applied Harris corner detectors to identify the interest area of detection as they assumed that vehicles usually contain a large number of edges and corners.

Point descriptor is also used in classification method apart from HoG which acts as an area descriptor. Sahli *et al.* [9] proposed a local feature-based approach based on Scale-Invariant Feature Transform (SIFT) [10]. They used

SIFT feature of vehicles and background to train a SVM classifier to create a model that was used to classify vehicles and background in query images. They obtained an accuracy of 95.2%. Comparing the detection results between the HoG feature and SIFT feature it apparently seems that SIFT feature is better. However, in terms of real-time detection, SIFT feature needs to use more computational resources especially when processing the whole image for small targets. In this paper, the proposed approach integrated feature based method and sliding window method by using HoG feature with a corner detection algorithm FAST (Features from Accelerated Segment Test) which can process quicker than the SIFT feature. Furthermore, the SIFT features have been applied in the tracking section because of its high matching accuracy and the long processing time problem has tackled by narrow the search area that the targets are most likely to appear in the tracking process.

In object tracking, various features are used such as points [11], models [12], shapes [13], and motions [14]. This paper focuses on the methods using object points and their motion. Window tracking is a widely used approach in object tracking [15]. The tracked object is described by a window template which can be a sub-image or a histogram feature etc. In the existing tracking approaches, the window tracking can be divided into static template model [16] and adaptive model [17]. The main difference between them is that the template will be updating during the process in the adaptive model and the other is not. However, one drawback of the window tracking is that its template has limited capabilities for modelling the appearances of the target. In this process, an adaptive discriminative tracking model has proposed which the model template of the targets are updated continually in both offline and during the process. The positive results in the neighbourhood frames by the tracking process are used to be the positive training samples in the following detection and tracking process, similarly, the negative results are used as negative training samples. The update strategy can handle the problems of changing appearance of the target and short-term occlusion which is another problem in tracking as tracking will be affected by any frames lost or random similar appearances of background during tracking.

Kalal *et al.* [4] proposed a Tracking-Learning-Detection (TLD) approach as the solution to long-term tracking which built an online feature detector from the first frame of a single tracking target. The detector continuously searches the target during entire tracking process and generates positive and negative samples that can update the detector for further tracking. TLD approach addresses the problem of recovering the tracking target in the event of tracking failures but it can only track the area selected in the first frame by the operator. Some vehicle tracking approaches have used colour-based particle filter features [18] which produce reasonable tracking results. However, this method is based on measuring the similarity of colour distribution between frames, making it likely to miss-track the target when a similar colour from background or other object occurs.

III. Vehicle Detection

A. Feature Density Estimation based on FAST

The first step in Fig. 1 is The FAST detection. The FAST detector developed by Rosten and Drummond [2] is based in principle on the SUSAN corner detector [19]. The FAST detector classifies a pixel $\boldsymbol{p}$ as a corner by performing a simple brightness test on a discretized circle of sixteen pixels around the pixel $\boldsymbol{p}$. A corner is detected at p if there are twelve contiguous pixels in the circle with intensities that are all brighter or darker than the centre pixel $\boldsymbol{p}$ by a threshold $\boldsymbol{t}$. A score function is evaluated for each candidate corner in order to perform non-maximal suppression for the final detection where $\boldsymbol{S_{bright}}$ is the subset of pixels in the circle that are brighter than $\boldsymbol{p}$ by the threshold $\boldsymbol{t}$,and S_{dark} the subset of pixels that are darker than $\boldsymbol{p}$ by $\boldsymbol{t}$.

$$\text{Score}\,(p) = \text{MAX}\left(\sum_{q\in S_{Bright}}|I_q - I_p| - t,\ \sum_{q\in S_{dark}}|I_p - I_q| - t\right) \quad (1)$$

Having detected the FAST corner, the next step is to select the interest regions where have high density of the corner that more likely have objects where the HoG will be applied.

B. Histogram of Oriented Gradients (HoG)

The HoG feature proposed by Dalal and Triggs [1] was originally developed for detecting humans. The idea of the HoG descriptor is that the shape of the objects can always be identified by the distribution of the edge even without precise information about the edges themselves. HoG can be well applied in vehicle detection because edges and shapes of the vehicles can generally be grouped into two major edge orientations. These edge orientations are largely perpendicular; therefore this gives a common distribution of edge directions among vehicles. In addition, the HoG descriptor has more advantages as it is relatively invariant to geometric and photometric changes. However, a weakness of the HoG descriptor is that it is not rotationally invariant. To solve this problem, four different directions (0, 45, 90, 135 degrees) of each training samples were used in the proposed method. Each group of the orientated training sample has its own classification model and the final classification model is calculated based on all four of the orientated classification models. The extraction of a HoG feature vector starts with colour and gamma normalisation, then edges are detected by convolving the image patch with the simple mask [-1, 0, 1] both horizontally and vertically. The image patch is then subdivided into rectangular regions cells, and within each cell the gradient for each pixel is computed. In the next step each pixel computes a weighted vote for the orientation of the cell by the gradient magnitude. Those votes are accumulated in to orientation bins with the range of 0 to 180 degrees which identify as the gradient angle that stored in a histogram. Local contrast normalisation is used to suppress the effects of changes in illumination and contrast with the background on the gradient magnitude. This step was found to be essential for better performance which is achieved by grouping cells into large blocks and

normalising within these blocks, ensuring that low contrast regions are stretched. Finally the normalised orientation histogram for each cell are collected together and result in a $b \times c_x \times c_y$ dimensional feature vector where b is the number of orientation bins and $c_x \times c_y$ is the number of image cells. The HoG feature vectors extracted from the regions of interest are imported into a binary classifier that determines the presence of a vehicle in the image patch. The method used separate SVMs to train on sample vehicle images that are categorised into four angular offsets (0, 45, 90, and 135). These four SVM's models are then intergraded as a single classifier model that evaluates a rotationally invariant response for a single HoG feature vector. The Support Vector Machines were chosen as the learning algorithm used in classification as they demonstrated a very high accuracy in previous vehicle detection research [7].

IV. Vehicle Tracking

The proposed tracking framework designed as: Tracker estimates the motion of vehicle or vehicles between the frame sequences. Detector processes in each frame independently and localise the target vehicle or vehicles based on the training classifier. The training classifier updates constantly from the learning process. The learning component also estimates the errors of the detector which it can make two types of errors: the false positive and false negative. In addition, the learning component also can generate positive and negative training samples based on the error estimation for the future detection to avoid errors. It is assumed that both detector and tracker can make errors so FBT has been proposed to monitoring the performance of the tracker. By using the proposed method, more training samples based on the current input video can be generated which the classifier will be updated more accurate.

A. Forward and Backward Tracking

This method followed the TLD tracking algorithm based on the optical flow and extended it to track multiple targets. The FBT method has been proposed to monitor the vehicle tracking results and the detection results. The FBT runs in parallel with the detection and monitor the tracking results by setting the detection results as ground truth. It can also run self-check based on the prior and after information.

An algorithmic description of FBT is given in Alg. 1. After the detection process, all detected vehicles have been labelled by the coordinates $C_n^f(x_n^f, y_n^f)$ and the image patches I_n^f where the n is the numbers of the vehicles and the f is the frame numbers and I_n^f also defined as the detector. Set $F_t$as an input frame image at time t and R_t^n as the region of interest (RoI) of the detected vehicle and n is the numbers of vehicles have been detected. In the FBT, the RoI are extracted from the vehicles' coordinates in the previous frame. In other word, the RoI are the predicted areas of the targets. The SIFT point matching process is conducted with the image patches of the targets I_n^f and the predicted regions R_{t+1}^n in the following frame to check if the targets is appeared in their corresponding predicted regions with their SIFT descriptors S_n^I and S_n^R. The set of sequent regions define the tracking results $T_t\{R_0R_1 \ldots R_t\}$ at the time t, by representing the positions of the target in each frame where T_t also identified as a tracker.

Algorithmically, the tracker $\boldsymbol{T_t}$ in the tracking system computes the similarity of the SIFT features between the detected vehicles' image patches and the RoI areas. The detected vehicles area based on the detection of first frame of the video, this area is considered as a sample of the vehicle that is used to find matched features in the regions of following frames and each frame of tracking will give a positive or negative results. However, these results might be inaccurate if there is any vehicle that is similar to the sample vehicle. For instance, it may have the matching results just above the threshold or when the target exits the image there is a vehicle similar to the target. The proposed a FBT method is to solve this kind of issue. It is assumed that if the tracker is tracking the same vehicle in the video the features of that vehicle will be highly similar to each other. Thus, this paper compared each tracking results forward to the next frame and backward to the previous frame.If the similarity of the feature is lower than the threshold it considered a lost target or a target that has left the image. The vectors of the system interact as follows. Regions of interest R_{t+1}^n are selected according to the vehicle positions C_n^f, the SIFT vector S_n^I and S_n^R are calculated for the R_{t+1}^n and the detected vehicles area I_n^f.

Algorithm 1 The Forward and Backward Tracking

Input: C_n^f ,I_n^f ,L
$R_{t+1}^n \leftarrow$ generateRoI(C_n^f)
for
$S_n^I \leftarrow$ SIFT (I_n^f)
$S_{n+2}^R \leftarrow$ SIFT (R_{t+1}^n)
End for
Match (S_{n+1}^I ,S_{n+2}^R) $\leftarrow$ Matching results M
If $M > L$
$T \in P$
else
$T \in N$

The SIFT vectors S_n^I and S_n^R make a match process to each other if the match results is higher than the threshold score L, then the tracker T^f will be addressed to the position of the targets and the target areas are considered as the positive results P. Otherwise the trackers T^fwhich has highest match but not reached to the threshold L are considered as the negative results N. The tracked vehicle databases M_P, M_N are created to store the positive results and the negative results, whose results will be use in further tracking and detection processes. The main purpose of creating the TVD M is to let the system continuously update the database for the detection classifier. For each frame the regions of interest does the matching process with each vehicle in positive memory M_P.

In the tracking process, SIFT point matching method was used and each tracked vehicle has its own combination of the SIFT points. Firstly, a tracker produces a trajectory of vehicle by tracking the SIFT points forward in time. Secondly, the point location in the last frame initializes a validation trajectory which is obtained by backward tracking from the last frame to the first one. Finally, the two trajectories are compared to each other and if there is a significant difference between them the tracking results will be recorded as an error. let $\boldsymbol{V} = (\boldsymbol{I_t}, \boldsymbol{I_{t+1}} \ldots \boldsymbol{I_{t+k}})$ be the sequence of frames and x_t be the trackers points location in time t. the points x_t is the tracked forward for k steps. The resulting trajectory is $T_f^k = (x_t, x_{t+1} \ldots x_{t+k})$ where f stands for forward and k indicates the length. The purpose is to estimate the error of the trajectory T_f^k given the frame image sequence S. The points x_{t+k} is tracked backward from current frame y_p to the first frame and produces $T_b^k = (x_t, x_{t+1} \ldots x_{t+k})$. The error is defined as the distance between these two trajectories (T_f^k and T_b^k). When the error occurs, the current tracking target will be considered as false negative.

The main advantage of this FBT is that it can prevent tracking fails in the case of the moving target being faster than expected, or when the target is temporarily blocked by the environment etc. Also with this method, multiple targets can be tracked in the video.

B. Self-Learning System

This section introduces the self-learning process. The purposed of the self-learning is to improve the performance of the vehicle detection. In each frame image the detector(s) will be evaluated for the error estimation. Two different self-learning systems are included in the proposed approach: positive inspectors P and negative inspectors N. Positive inspectors are used to identify whether the tracker labelled as positive by the classifier have been recognised as negative by the detector. Negative inspectors are used to identity whether that trackers labelled as negative by the classifier have been recognised as positive by the detector.

In the tracking process, let T be a result from the tracker and L be a label from $Y = \{-1,1\}$. A set of testing frames I is considered as unlabeled data and the $S = \{(T, L)\}$ is considered as labelled data. The input to the self-learning is a labelled dataset S_l and an unlabeled dataset S_u where $l \ll u$. The task of the self-learning process is to learn and update the detection classifier f from the labelled tracking results S_l and bootstrap its performance by the new detection results. The detection classifier f is corresponds to the estimation of targets from the training set with the TVD. However, there is an exception where is the TVD is iteratively augmented by the initial training samples created for the detections. The training process is initialized by inserting the tracking results to the classification set and by estimating the classifier parameters θ. As mentioned that the process proceeds iteratively and in the iteration k, the classifier trained in $k-1$ assigns labels to the training samples formed from the tracking results, $y_T^k = f(T|\theta^{n-1})$ for all $T_n \in I_n$. Note that, the classifier can operate on multiple trackers at same time. Then self-learning system is used to verify that the labels assigned by the classifier are correct or not. The samples labels that violate the process are corrected and added to the TVD. The iteration is finished by retraining the classifier with the updated tracking results.

The positive inspectors samples are then inserted into the TVD thus improve the generalization properties of the classifier. Negative are used to identify the trackers that have been labelled as negative by the classifier but the detector labelled as positive. The negative samples then extend the pool of the TVD which improve its discriminative properties of the classifier. Based on the proposed theory, the errors between the tracker and the detector have been analysed. In the classifier f, the errors will be characterised by false positives α (f) and false negatives β (f). Let the $n_c^+(f)$ be the number of training samples for which the label was correctly changed in the TVD and $n_i^-(f)$ is the number of samples for the labels that was incorrectly changed in the TVD. The error of the classifier will be:

$$\alpha(\mathrm{K}+1) = \alpha(k) - n_c^-(f) + n_i^+(f) \quad (2)$$

$$\beta(\mathrm{K}+1) = \beta(k) - n_c^+(f) + n_i^-(f) \quad (3)$$

The quality of the checking process is characterized by four measures. P-true is the number of correct positive samples divided by total number of results. P-false is the number of correct positive samples divided by the number of false negatives; N-true is the number of correct negative samples divided by number of results. Finally the N-false is the number of correct negative samples divided by the total number of false positives. It is assumed here that the self-learning are characterised by fixed measure throughout the training. The number of correct and incorrect results then expressed as follows:

$$n_c^+(K) = R^+ \beta(f),\ n_i^+ \frac{(1-P^+)}{P^+} R^+\beta(f) \quad (4)$$

$$n_c^-(K) = R^- \alpha(f),\ n_i^- \frac{(1-P^-)}{P^-} R^-\alpha(f) \quad (5)$$

In the real-time, it might be not possible to identify all the error between the detectors and trackers. Therefore, the training cannot converge to all the tracking results. There is another scenario that assuming if the detection result is incorrect which can leads the tracking process track the wrong target. Also in the tracking process, occlusion issue is a key problem that causes tacking error. These problems can be tackled by the proposed FBT process.

V. Experiments and Resutls

This paper used 5 different videos under certain circumstances to test the proposed system. They are aerial videos captured by UAV to detect vehicles on a highway. The five videos contain different scenarios including 1) Complex background, 2) Vehicle occlusion, 3) Blocked vehicles, 4) Blurred vehicles and 5) Changing vehicle size and appearance. This paper compared the proposed

approach with several other different methods in vehicle detection in the experiment including HoG [8], SIFT [10] and PLS Hough [20]. It also compared with tracking algorithms including original TLD [4], STRUCK [21] and CoGD [22]

A. Detection Results

To test the performance of the detection, the detection rate on correctly identifying regions that contain vehicles in the entire testing videos have been evaluated. Assuming that for each frame ***t*** the number of positive detections is indicted by P_t and the number of ground-truth vehicles is indicted as $N_G^{(t)}$, the detection accuracy is calculated as:

$$\text{Detection accuracy} = \frac{\sum_{t=1}^{N_{frames}} P_t}{\sum_{t=1}^{N_{frames}} N_G^{(t)}} \quad (6)$$

TABLE I. DETECTION RESULTS

video	*HoG*	*SIFT*	*PLS Hough*	*SLTD*
Video 1	89.90%	70.14%	70.52%	**94.06%**
Video 2	80.17%	70.96%	74.21%	**91.86%**
Video 3	73.43%	74.18%	67.15%	**97.27%**
Video 4	82.21%	93.01%	75.33%	**97.03%**
Video 5	75.09%	96.05%	97.91%	**94.06%**
Avearage	80.16%	80.87%	77.02%	**94.86%**

The Table I shows the comparison of vehicle detection results using several methods. The result shows that the proposed system obtained better detection accuracy than others: **0.9486** across all the video dataset.

B. Tracking Results

To perform this tracking experiment, the same testing videos as used in the previous vehicle detection experiment have been used. For each frame, the number of correctly tracked vehicles was considered as True Positives (TP), the number of background regions that were incorrectly classified as vehicles was considered as False Positives (FP), and the number of vehicles that were missed in detection was considered as False Negatives (FN) were recorded. The tracking results are calculated by:

$$F_1 = 2 \times \frac{PR}{P+R} \quad (7)$$

where F_1 is a harmonic mean of precision P and recall R. F_1 gives a value in the interval (0, 1) with a larger value corresponding to a higher classification rate. The Multiple Object Tracking Accuracy (MOTA) metric propose [10] was used to measure an accuracy score that considers the number of missed detection, the false positive rate and the mismatches between tracker and the vehicles. Table II shows the tracking comparison results. Note that the tracking comparison only evaluated the tracking performance on single vehicle in TLD, this is because the TLD can only track single target and other methods used 10 vehicles for tracking test. The result shows that the SLTD achieved the best score in each video and matched the performance of the original TLD for single vehicle tracking while SLTD has the advantage of tracking multiple targets.

TABLE II. TRACKING RESULTS

Video	*CoGD*	*STRUCK*	*TLD*	*SLTD*
Video 1	1.00/1.00/0.97	0.23/0.23/0.57	0.94/0.94/N/A	**0.97/0.98/0.99**
Video 2	1.00/0.99/0.98	1.00/1.00/0.95	0.86/0.77/N/A	**0.89/0.83/0.90**
Video 3	1.00/1.00/0.99	0.84/0.84/0.88	1.00/0.95/N/A	**1.00/1.00/0.98**
Video 4	0.72/0.92/0.88	0.26/0.21/0.37	1.00/0.94/N/A	**1.00/1.00/0.99**
Video 5	0.85/1.00/0.93	0.88/0.88/0.90	0.93/0.83/N/A	**0.95/1.00/0.95**

The performance measured by Precision/Recall/MOTA

VI. CONCLUSIONS

This paper proposed a Self-Learning Tracking Detection method for vehicle detection and tracking from UAV video. This method has solved a problem in vehicle detection and tracking where the training samples are taken from other vehicles that are different from the vehicles in testing. The appearance of the vehicle in the captured video is affected be the attitude, speed and the image resolutions of UAVs. The proposed method can learn the vehicle features and created a unique detection model for each vehicle during the tracking process. A Forward and Backward Tracking mechanism was proposed to check the errors from the tracking and detection process. The proposed method demonstrated a reasonably high accuracy and can successfully detect and track a variety of differing vehicle types under varying rotation, sheering and blurring conditions. Notice that the proposed method in this paper is inspired by the TLD algorithm and the compared tracking results have shown that the proposed method can achieve higher tracking accuracy than the TLD under some complex circumstances such as occlusions and blocked vehicles in a complex background. This paper also compared the proposed approach with other tracking and detection approaches by using 5 different videos captured from UAVs in different situations. The results show that the proposed approach has slightly better performance. For the future work a larger and more diverse datasets with more varieties of vehicles and background will be used to train the system in order to improve the classifier for initial detection in order to perform better learning model

in the tracking process. Also, the lerning component could be improved by revising the error checking part between the detector and tracker.

REFERENCES

[1] D. Navneet and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 886-893.

[2] M. Trajković and M. Hedley, "Fast corner detection," *Image and Vision Computing*, vol. 16, no. 2, pp. 75-87, 1998.

[3] C. Corinna and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[4] K. Zdenek, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409-1422, 2012.

[5] M. Cheng, N. J. Mitra, X. M. Huang, P. H. S. Torr, and S. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569-582, 2015.

[6] A. Anelia, A. Krizhevsky, and V. Vanhoucke, "Pedestrian detection with a large-field-of-view deep network," in *Proc. IEEE International Conference on Robotics and Automation*, 2015, pp. 704-711.

[7] X. C. Shi, H. B. Ling, E. Blasch, and W. M. Hu, "Context-driven moving vehicle detection in wide area motion imagery," in *Proc. 21st International Conference on Pattern Recognition*, 2012, pp. 2512-2515.

[8] G. Joshua, A. V. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis, "Vehicle detection from aerial imagery," in *Proc. IEEE International Conference on Robotics and Automation*, 2011, pp. 2065-2070.

[9] S. Samir, Y. Ouyang, Y. L. Sheng, and D. A. Lavigne, "Robust vehicle detection in low-resolution aerial imagery," *Proceedings of SPIE*, pp. 76680G-76680G, 2010.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[11] H. L. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 297-309, 2015.

[12] K. H. Lee, J. N. Hwang, and S. I. Chen, "Model-Based vehicle localization based on 3-D constrained multiple-kernel tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 1, pp. 38-502015.

[13] W. C. Wu, E. A. Bernal, R. P. Loce, and M. E. Hoover, "Multi-resolution video analysis and key feature preserving video reduction strategy for (real-time) vehicle tracking and speed enforcement systems," U.S. Patent 8,953,044, February 10, 2015.

[14] S. T. Jeng and L. Y. Chu, "Tracking heavy vehicles based on weigh-in-motion and inductive loop signature technologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 632-641, 2015.

[15] T. Carlo and T. Kanade, "Detection and tracking of point features," Pittsburgh: School of Computer Science, Carnegie Mellon Univ., 1991.

[16] B. Sebastiano, G. M. Farinella, A. Furnari, G. Puglisi, A. Snijders, and J. Spiekstra, "An integrated system for vehicle tracking and classification," *Expert Systems with Applications*, 2015.

[17] N. Zhao, Y. J. Xia, C. Xu, X. M. Shi, and Y. C. Liu, "APPOS: An adaptive partial occlusion segmentation method for multiple vehicles tracking," *Journal of Visual Communication and Image Representation*, 2015.

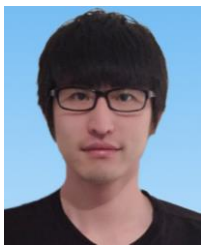
[18] X. F. Lu and L. Wang, "Vehicle tracking process based on combination of SURF and color feature," in *Proc. Sixth International Conference on Graphic and Image Processing*, 2015, pp. 94430W-94430W.

[19] R. Edward and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV*, Springer Berlin Heidelberg, 2006, pp. 430-443.

[20] T. Remma, K. Kato, D. Harwood, and L. S. Davis, "Vehicle detection using PLS Hough transform," in *Proc. 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision*, 2015, pp. 1-6.

[21] H. Sam, A. Saffari, and P. HS Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE International Conference on Computer Vision*, 2011, pp. 263-270.

[22] Q. Yu, T. B. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Computer Vision–ECCV*, Springer Berlin Heidelberg 2008, pp. 678-691.



Xiyan Chen received his Master Degree in Computer Science from Loughborough University. He is currently PhD student at the Loughborough University and his research area is the imaging process of vehicle detection and tracking in aerial videos.

Qinggang Meng is a Senior Lecturer in the Department of Computer Science, Loughborough University. Before joining Loughborough University, He worked as a postdoctoral research associate for 4 years in University of Wales, Aberystwyth (UWA), UK. He obtained his PhD from the Department of Computer Science, UWA. Before he came to UK, he did one year RA in City University of Hong Kong on intelligent grasping control. Before that, he was working in Intelligent Machine Institute at Tianjin University for several years in the area of intelligent robotics.