Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 2, February 2019, pp. 688-699
DOI: 10.23940/ijpe.19.02.p33.688699

Vehicle Detection on Unmanned Aerial Vehicle Images based on Saliency Region Detection

Wenhui Li[a], Feng Qu[a,*], and Peixun Liu[b]

[a]*College of Computer Science and Technology, Jilin University, Changchun, 130021, China*
[b]*Changchun Institute of Optics, Fine Mechanics and Physics*
Chinese Academy of Sciences, Changchun, 130033, China

Abstract

The target detection and tracking technology of the unmanned aerial vehicle (UAV) is an important research direction in the field of UAV aerial photography. In order to effectively and accurately detect vehicles in a UAV platform and in complicated road environments, the authors proposed a vehicle detection method based on saliency region detection. First, the saliency map of the target is calculated by using the salient region detection method based on the optimized frequency-turned. Next, segmentation methods based on Boolean Map and OTSU are combined to determine the region of interest of the vehicle target in the saliency map image. Finally, a series of vehicle apparent features-based methods based on geometry, symmetry, and horizontal edge wave are used to determine the vehicle and eliminate the interference of roadside objects accurately. Experimental tests carried out from different datasets show excellent performance in multi-vehicle detection in terms of accuracy in complex traffic situations and under different scales and angles of aerial images, realizing fast vehicle detection on the UAV platform.

Keywords: unmanned aerial vehicle; free motion camera; vehicle detection; salient region detection

(Submitted on November 11, 2018; Revised on December 15, 2018; Accepted on January 6, 2019)

1. Introduction

The Unmanned Aerial Vehicle (UAV) is an aircraft that has capability of being remotely piloted, autonomous, semi-autonomous, or fully automatic flying. UAVs are mainly divided into fixed-wing UAVs, helicopters, and multi-rotor UAVs. Prior to 2010, fixed-wing drones and unmanned helicopters dominated the aerial photography field. However, in recent years, with the rapid development of control technology, sensor technology, and computer vision, multi-rotor UAVs have become increasingly popular in the field of aerial photography.

Target detection and tracking technology is a key technology in the field of computer vision, and it has attracted the attention of scholars. In the fields of drone aerial photography, target detection and tracking are indispensable for tracking shooting [1]. Therefore, target detection and tracking technology is an important research direction in the field of UAV aerial photography. In addition, the target in the aerial video is small and the background is complex. Target detection is easily affected by interferences of scale changing, rotation, light variation, occlusion, and camera shaking, which makes the detection and tracking of targets in aerial video more difficult.

The accuracy of the target detection will directly affect the subsequent processing of the target location. Therefore, the target detection technology plays a crucial role in the target detection and tracking system based on the UAV aerial video. At present, the moving object detection algorithm is relatively mature, and the target detection algorithms that can be applied to drone aerial video mainly include the following categories:

(1) Target detection method based on frame difference [2-4]. The moving target is determined mainly by grayscale value difference between pixels in the same position of two successive frames. [2] applied a background subtraction method

* Corresponding author.
E-mail address: qufeng_jlu@163.com

based on median to detect vehicles. [3] applied a frame difference method, combining with the image registration process to detect moving vehicles. The frame difference-based algorithm is simple to operate and easy to implement; however, this algorithm is only suitable for the target detection in static background. In other words, it is only applicable to the detection of moving targets in the hovering state of a UAV, and the application range is limited.

(2) Target detection method based on optical flow [5]. The optical flow-based method uses the optical flow as the instantaneous field of grayscale pixel in the image to achieve the target detection. The optical flow-based method is mainly divided into four categories according to the principle: a gradient-based method, a matching-based method, an energy-based method, and a phase-based method. However, optical flow methods are sensitive to background motions.

(3) Target detection method based on feature matching. The target template is established by extracting the features of the target to be detected (corner features, colour features, etc.), and then the similarity of the target template in the real-time video is extracted to detect the target. At present, the most commonly used feature matching algorithms are SIFT [6], SURF[7], BRISK [8], and FREAK [9]. Feature matching is the most widely used target detection and recognition algorithm, and the feature matching algorithm is suitable for both the target detection of dynamic background and the detection of static targets [10].

(4) Target detection algorithms based on machine learning. In recent years, object detection algorithms, such as bLPS-HOG+SVM [11], V-J+SVM [12], HOG+SVM [13], Disparity Maps + HOG based detector [14], discriminatively trained deformable part model (DPM) [15], and V-J object detection scheme [16], have become popular for vehicle detection in UAV videos. Object detection algorithms are less sensitive to image noise, background motions, and scene complexity. Therefore, they are more robust [17]. However, due to the computational complexity, the machine learning-based method is slow in detecting multiple objects and therefore cannot satisfy the requirement of real-time applications.

In summary, this research aims to propose a method that can detect vehicles from UAV videos quickly and accurately. The rest of our work is organized as follows: a fast vehicle ROI detection algorithm based on modified frequency-tuned saliency region detection is proposed in Section 2. Section 3 briefly introduces a target region of interest segmentation method in our work. Then, a vehicle verification algorithm based on horizontal edge wave is further presented in Section 4. Section 5 shows a comprehensive evaluation of the proposed method using different scenarios. Section 6 finally concludes this paper.

2. Fast Vehicle ROI Detection Algorithm based on Modified Frequency-Tuned Saliency Region Detection

Through the analysis of the test videos and vehicle samples under a large number of UAV aerial images, we found that vehicles have significant visual salient features with respect to the road surface and most of the background area. In addition, the results of our previous work [18] show that the pixel grayscale values in the shadow area at the bottom of the vehicle basically all fall within the first 5% of the grayscale histogram of entire image. However, only the vehicle bottom shadow region can be detected by using the shadow detection method. Using the visual saliency detection method can not only detect the vehicle region, but also detect the shadow region under the vehicle in the image; this can increase the probability of extracting the vehicle region of interest [19]. Therefore, we first use the visual saliency detection method to determine the region of interest of the vehicle in the aerial image. At present, the target detection method based on saliency detection is a research hotspot in the field of target detection.

In recent years, a large number of saliency detection methods have emerged. Figure 1 illustrates comparison results between state-of-the-art saliency detection algorithms that are processed on the data set in the experiment section of this paper: (a) is the original image, (b) is the grayscale image of (a), (c) is the ground truth, (d-n) are the respective saliency map results of AC [20], RC [21], HC [22], LC [23], IM [24], SUN and SUN-CON [25], IT [26], SaliencyMap [27], and FT-LAB and FT-L [28], which are processed on Frame #70 of testvideo1 (video resolution: 960×540) in our data set. Experimental results show that the AC, RC, LC, HC, and IT methods can quickly detect the target region in the image. However, as can be seen from Figure 1 (d), (e), (f), and (g), except for the target, the saliency eigenvalues are relatively close to a large number of interference regions; therefore, the subsequent segmentation algorithm will split out more interference regions. It can be seen from Figure 1 (h), (i), and (j) that the detection results of IM, SUN, and SUN-CON are quite satisfactory; however, the processing times of these three algorithms all exceed 2900ms. The algorithm has high complexity and cannot be applied to real-time detection system. The principle of the Saliency Map and FT-LAB are roughly the same; however, since FT-LAB performs Gaussian smoothing on the L, A, and B components of the LAB color space of the image, the complexity of the algorithm is relatively high. If the FT-LAB is reduced and only the L channel is processed, the processing result of the algorithm is shown in Figure 1 (a). It is almost the same as FT-LAB, and the processing speed of the algorithm is shortened by half. Although the processing speed is improved, the processing speed of 858 ms/frame still

bcannot meet the engineering requirements. Therefore, inspired by the FT-LAB algorithm, a fast region-of-interest extraction method based on saliency region detection is presented in this paper. The method includes two main parts: the method of fast saliency map generation and the efficient method of target region segmentation. FT-LAB is first improved to adapt to embedded system-based saliency detection to detect the vehicle's area of interest under UAV aerial video better.

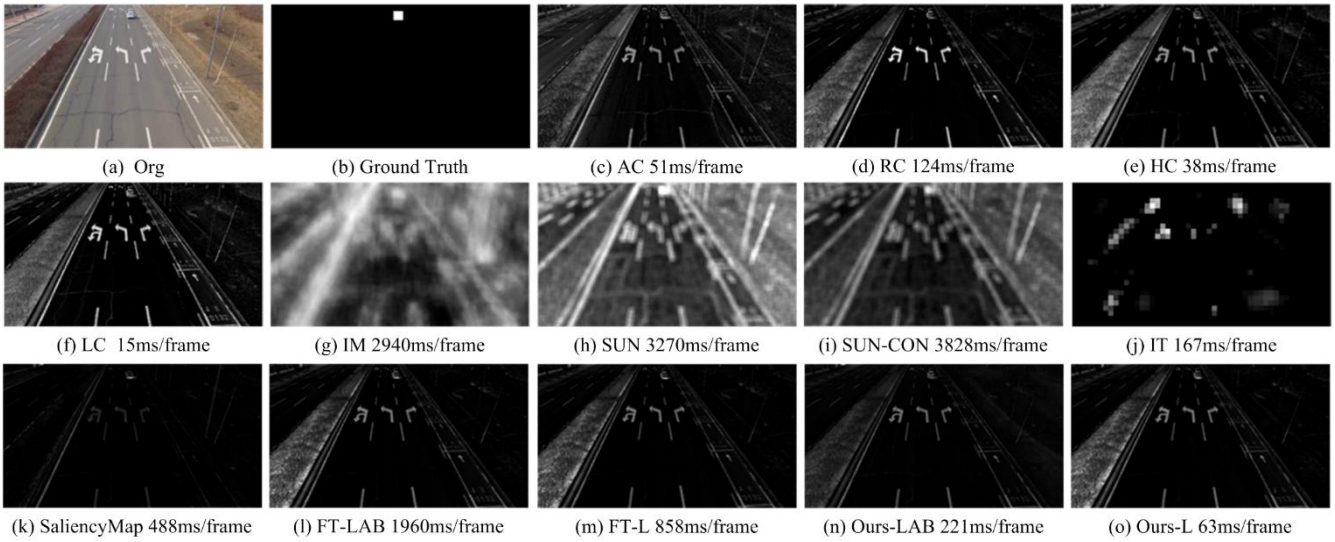

Figure 1. Comparisons between state-of-the-art saliency detection algorithms

In this section, we detail how to improve the frequency-tuned saliency region detection to make it more suitable for fast saliency region detection. The processing effects and running time of the algorithm based on three components of LAB and *L* component are obtained respectively. The general flow of the method is to first convert the image from RGB color space to Lab color space using the following formula and then generate RGB2Lab look-up tables, which are defined as *Lf*[*r*][*g*][*b*], *af*[*r*][*g*][*b*], and *bf*[*r*][*g*][*b*], where *r*, *g*, and *b* are three components of pixels in RGB color space.

$$\begin{cases} R = gamma(\dfrac{r}{255.0}) \\ G = gamma(\dfrac{g}{255.0}) \\ B = gamma(\dfrac{b}{255.0}) \end{cases} \tag{1}$$

$$gamma(x) = \begin{cases} (\dfrac{x + 0.055}{1.055})^{2.4}, & x > 0.04045 \\ \dfrac{x}{12.92}, & \text{otherwise} \end{cases} \tag{2}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}, M = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \tag{3}$$

$$f(t) = \begin{cases} t^{\frac{1}{3}}, & \text{if } t > (\dfrac{6}{29}) \\ \dfrac{1}{3}\left(\dfrac{29^2}{6}\right)t + \dfrac{4}{29}, & \text{otherwise} \end{cases} \tag{4}$$

$$\begin{cases} L = 116f\left(\dfrac{Y}{Y_n}\right) - 16 \\ a = 500[f\left(\dfrac{X}{X_n}\right) - f(\dfrac{Y/}{Y\_n})] \\ b = 200[f\left(\dfrac{Y}{Y_n}\right) - f\left(\dfrac{Z}{Z_n}\right)] \end{cases} \tag{5}$$

Here, the *L*, *a*, and *b* look-up tables are shown as Equations (6)-(8) and can be calculated from Equations (1)-(5).

$$Lf[r][g][b] = \begin{bmatrix} Lf[0][0][0] = 0.0 \\ Lf[0][0][1] = 0.019789 \\ \vdots \\ Lf[255][255][254] = 99.975167 \\ Lf[255][255][255] = 100.000004 \end{bmatrix} \tag{6}$$

$$af[r][g][b] = \begin{bmatrix} af[0][0][0] = 0.0 \\ af[0][0][1] = 0.139059 \\ \vdots \\ af[255][255][254] = 0.172195 \\ af[255][255][255] = 0.002438 \end{bmatrix} \tag{7}$$

$$bf[r][g][b] = \begin{bmatrix} bf[0][0][0] = 0.000000 \\ bf[0][0][1] = 0.378486 \\ \vdots \\ bf[255][255][254] = 0.471613 \\ bf[255][255][255] = 0.004647 \end{bmatrix} \tag{8}$$

If the input image is a grayscale image, Equations (6)-(8) can be simplified to the lookup table of the *L* component, which is shown in Equation (9).

$$Lf[gray] = \begin{bmatrix} Lf[0][0][0] = 0.0 \\ Lf[0][0][1] = 3.542353 \\ \vdots \\ Lf[255][255][254] = 99.848167 \\ Lf[255][255][255] = 100.0 \end{bmatrix} \tag{9}$$

The average of the *L*, *a*, and *b* components of the whole image *lavg*, *aavg*, and *bavg* are calculated firstly, and then *slvec*, *savec*, and *sbvec* are obtained by filtering the *L*, *a*, and *b* components using the difference of Gaussian (DoG) filter proposed in [28].

$$DoG(i,j) = \frac{1}{2\pi}\left[\frac{1}{\sigma_1^2}e^{-\frac{(i^2+j^2)}{2\sigma_1^2}} - \frac{1}{\sigma_2^2}e^{-\frac{(i^2+j^2)}{2\sigma_2^2}}\right] = G(i,j,\sigma_1) - G(i,j,\sigma_2) \tag{10}$$

Where $\sigma_1, \sigma_2 (\sigma_1 > \sigma_2)$ are the standard deviations of the Gaussian. A *DoG* filter is a simple band-pass filter whose passband width is controlled by the ratio $\sigma_1 : \sigma_2$. Let us consider combining several narrow band-pass *DoG* filters. If we define $\sigma_1 = \rho\sigma$ and $\sigma_2 = \sigma$ such that $\rho = \sigma_1 : \sigma_2$, we find that a summation over *DoG* with standard deviations in the ratio ρ results in:

$$\sum_{k=0}^{K-1} G(i,j,\rho^{k+1}\sigma) - G(i,j,\rho^k\sigma) = G(i,j,\sigma\rho^K) - G(i,j,\sigma) \tag{11}$$

For an integer $K \geq 0$, it is simply the difference of two Gaussians whose standard deviations can have any ratio $N = \rho^K$. We follow the parameter selection section of [28] to set the parameter of the *DoG* filter. Inspired by [28], the saliency map *Sap* for a grayscale image *I* can be formulated as:

$$Sap(i,j) = \|lavg - slvec(i,j)\| \tag{12}$$

Where *lavg* is the mean value of *L* component of the whole image *I*, *slvec(i,j)* is the Gaussian blurred version of *L* component of the pixel (*i*, *j*) to eliminate fine texture details as well as noise and coding artifacts, and $\|\cdot\|$ is the L_2 norm, which is the Euclidean distance. To extend Equation (12) to use features of color and luminance, we modified it as:

$$Sap(i,j) = ||\boldsymbol{I}_{avg} - \boldsymbol{I}svec(i,j)|| \tag{13}$$

Where $\mathbf{I}_{avg}$ is the mean image feature vector of Lab color space and $\boldsymbol{I}svec(i,j)$ is the corresponding image pixel vector value in the Gaussian blurred version of the image. Here, we used a 5×5 separate binomial kernel, and each pixel location is an $[L, a, b]^T$ vector using the Lab color space.

In Figure 2, (a) is the RGB test image of No. 5108 in the public test set, with a picture resolution of 400×300. (b) is the saliency map image of the FT algorithm, and the processing time is 452.17ms/frame. (c) is the result of our embedment of FT algorithm code [29]. The main modification is to change the vector variables in the original program to unsigned char * and double *, etc. We can see that the processing speed reached 50.76ms/frame shortened nearly ten times. (d) is the result of processing using a look-up table generated using Equations (6), (7), and (8). The processing speed of the algorithm is reduced to 34.21ms/frame, which basically achieves the real-time processing under the condition of ensuring that the processing result of the algorithm is consistent with the original FT. In order to verify the processing effect of algorithms on the grayscale image, we also use the grayscale image data set to test the algorithms. (f) shows the saliency map result after reducing the algorithm in [28]; only the L component processed, and the processing speed of the algorithm is 177ms/frame. (g) is the saliency map result if only the L component is processed by our method, and the processing speed is reduced to 13.12 ms/frame. (h) is the result of using the look-up table generated by Equation (9), and the algorithm achieves a processing speed of 7.7ms/frame while ensuring the same processing result as FT.

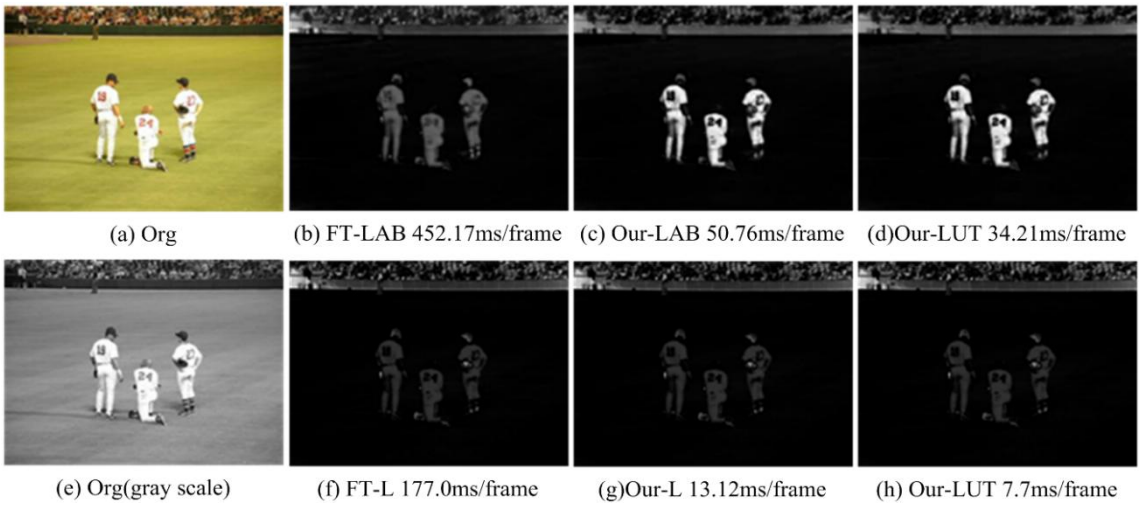

Figure 2. Saliency region detection results comparison between [28] and our methods

3. Target Region of Interest Segmentation Method

After the saliency map is extracted, a target segmentation method is needed to determine the position of the target in the image. The simplest and most common method is the threshold segmentation method, that is, set a fixed threshold. According to [28], the threshold is set to 100 for binary image segmentation. The method has the advantages of simple implementation and fast running speed. However, this method relies too much on prior knowledge and is not adaptive. The basic principle of the target segmentation algorithm based on the Boolean maps [27] is to set several thresholds of the threshold segmentation method according to a certain interval. The interval set in [27] is 8, that is, the segmentation threshold is respectively set as 8, 16, 24, ⋯, 256. Then, a series of binary images are obtained to form a pool of target regions of interest. Then, according to the principle that the target's grayscale value is more stable than the background's, the number of times that the target appears in a certain position in the image is greater than a certain threshold to determine the target regions, and it is finally combined with OTSU [30] to determine the final target area. The method can extract the target of simple background effectively, but because of the need to deal with a large number of different binary images, the method of processing speed is slow; processing a frame takes about 27s (resolution 960×540).

An adaptive threshold target segmentation method based on mean-shift is proposed by [28]. The mean-shift segmentation algorithm that is performed in Lab color space provides better segmentation boundaries. Instead of applying a fixed threshold, an adaptive threshold that is image saliency dependent is also presented in their method to segment each image in their database.

Otsu's Method is optimal and widely used for separating the target from its surrounds by computing a threshold maximizing the between-class variance. Experimental results show the method can separate the target well by using the

threshold that is formed by OTSU's result value for each image in every video sequence adding or subtracting a fixed value Δ.

As shown in Figure 3, (a) is the result of Boolean maps. (b) is the result of mean-shift combined with setting segmentation method threshold to two times the mean saliency of a given image. (c) is the result of original OTSU. (d), (e), and (f) are the results of [31] with different segmentation thresholds, which are set to 2, 4, and 6 times the mean saliency respectively. (g), (h), and (i) are the results from the OTSU method combined with Hou's method using different thresholds.

In our method, the segmentation results of 2*Avg, OTSU and OTSU + 2*Avg are used to generate a pool *P* of regions, which is generated as follows: the Sobel filter is firstly used to detect the edge of the binary images of four segmentation methods illustrated in Figure 3, and then the binary and edge images are used as the input of connected domain extracting method to generate a set of target regions of interest (Rois)*r*. A struct of *Region_t* is defined for later computations, and the elements in struct are as follows:

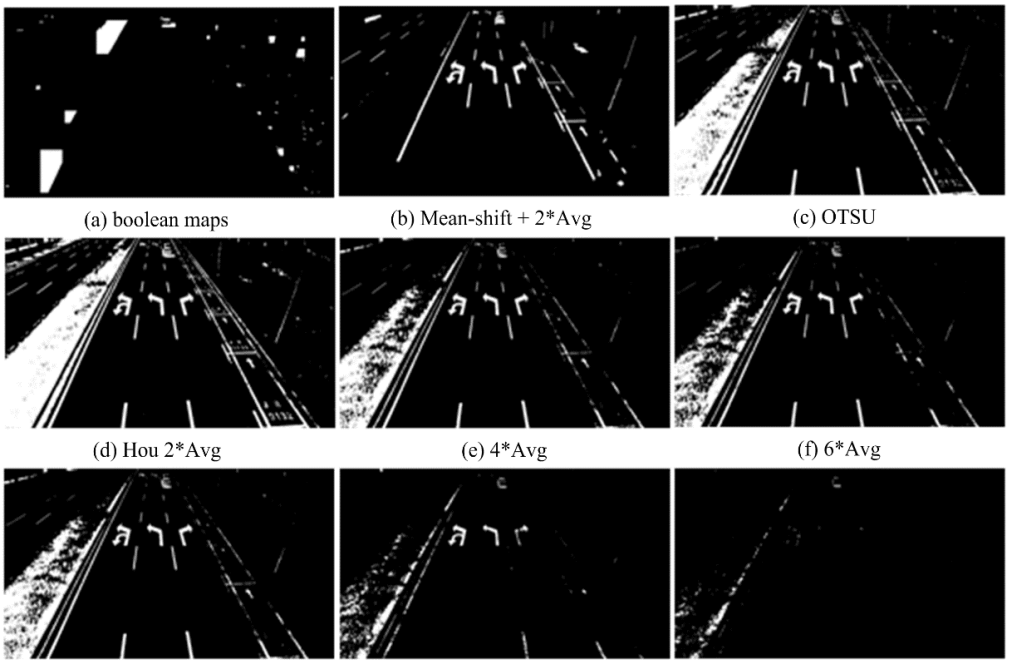

Figure 3. Comparison between target region of interest segmentation methods

wid, ***hei***, ***cx***, and ***cy*** are the width, height, x coordinate of the center, and y coordinate of the center of the bounding box respectively. The bounding box is calculated by connected domain extracting method.

fill rate is defined as $|r|/(wid \times hei)$, where $|r| = \sum_r I(x, y)$ indicates the number of pixels whose grayscale value is 255 in region *r*.

aspect ratio is the ratio of width to height, which is calculated by *wid/hei*.

symmetry is a statistic to describe the symmetry of vehicles that are obviously symmetrical objects. According to the symmetry-based method described in our previous work [32], the symmetry measure method based on normalized entropy is applied to calculate the symmetry value of ROIs. The symmetry is described as Equation (14):

$$sym = \frac{[\frac{S(x_s)+1}{2} + \frac{E(l)}{E_m}]}{2} = \frac{S(x_s) \times E_m + 2 \times E(l) + E_m}{4 \times E_m} \tag{14}$$

Where $S(x_s)$ is the symmetry value of the target. $E(l)$ is the information entropy, which is also the mathematical expectation of information content. E_m is the max value of information entropy. Inspired by [27], five definitions used for measuring region similarity between two labelled regions *i* and *j* are defined as follows:

$$\textbf{\textit{area variation}}: V_a(i,j) = |wid_i \times hei_i - wid_j \times hei_j| \tag{15}$$

$$\textbf{\textit{center distance}}: D_c(i,j) = \left\|c_i - c_j\right\|^2, \text{ where } \| \cdot \| \text{ is } l_2\text{-norm.} \tag{16}$$

$$\textbf{\textit{fillrate difference}}: D_f(i,j) = \frac{max\ (f_i,\ f_j)}{min\ (f_i,\ f_j)} \tag{17}$$

$$\textbf{\textit{aspect ratio difference}}: D_{ar}(i,j) = \frac{max\ (ar_i,\ ar_j)}{min\ (ar_i,\ ar_j)} \tag{18}$$

$$\textbf{\textit{symmetry difference}}: D_s(i,j) = \frac{max\ (s_i,\ s_j)}{min\ (s_i,\ s_{j)}} \tag{19}$$

Taking account of target has a set of similar segmentation results in the four different binary images that were previously introduced. Supposing the pool P is formed by $nRois$ $\{r_1, r_2, \cdots, r_n\}$, a clustering technique based on the spatial relationships and similarity between the regions is applied to cluster the regions belonging to the same target. *Rois* in the pool P that belong to the same cluster S_k are firstly verified by Equation (20).

$$D_c(i,j) \le \frac{min^2(wid_i, wid_j) + min^2(hei_i, hei_j)}{4} \tag{20}$$

Typically, there are several *Rois* in each cluster. Equation (21) is used to find the accurate target representation *ATR*, which is the largest region in the three pairs of regions (r_m, r_n), (r_p, r_q), and (r_x, r_y) belonging to the cluster S_k. The results of the target region of interest segmentation are shown in Figure 4.

$$ATR = arg\ max_{atr\in\{r_m,r_n,r_p,r_q,r_x,r_y\}}|r|, \text{ where } \begin{cases} (r_m, r_n) = argmin_{r_m,r_n\in S_k}\ D_f(r_m, r_n) \\ (r_p, r_q) = argmin_{r_p,r_q\in S_k}\ D_f(r_p, r_q) \\ (r_x, r_y) = argmin_{r_x,r_y\in S_k}\ D_f(r_x, r_y) \end{cases} \tag{21}$$

(a)Original Image (the 70th frame of ‘Test video 1’) (b) ROI Segmentation Results

Figure 4. Results of target region of interest segmentation method

4. Vehicle Verification based on Horizontal Edge Wave

Although the methods described in Sections 2 and 3 perform well in vehicle detection, other objects such as trees, lane lines, and traffic signs are sometimes falsely detected as vehicles. Figure 5(a), (f) illustrates vehicles and false alarms detected by our target region of interest segmentation method respectively. In order to reduce these influences, in this section a vehicle verification algorithm is introduced. It can distinguish vehicles and false alarms effectively by taking into account the horizontal edge wave feature that we defined to describe man-made objects. We first use the Sobel operator-based edge detection algorithm to extract the horizontal edge of the *Rois*, which are shown in Figure 5(b) and (g). A horizontal edge pixel histogram (*HEPH*) is generated by summing edge pixels in each column, and then horizontal edge waves that are

illustrated as Figure 5(c) and (h) are obtained by processing a Median filtering on *HEPH*. Experimental results show that vehicles have plenty of long horizontal edges (more than ten pixels in the same row). Figure 5(d), (i) are long horizontal edges of *Rois*, which are generated by using a filter on horizontal edge images. Figure 5(e), (j) shows horizontal edge waves of (d), (i) respectively.

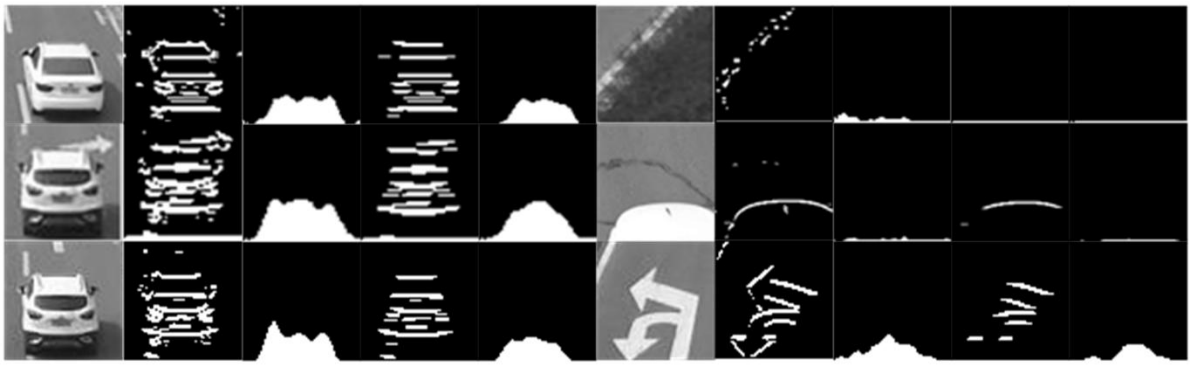

Figure 5. Vehicle (long) horizontal edges and their corresponding vehicle waves of Rois

After vehicle waves are generated, final bounding boxes of the vehicle can be accurately obtained by segmenting vehicle horizontal waves as Algorithm 1. The results are shown in Figure 6.

Algorithm 1. Accurate bounding box generation method based on vehicle waves

Step 1. Generate threshold $Threshold_{horWav}$.

The *x*-coordinates of the left and right borders of vehicles in the image are obtained by segmenting the vehicle horizontal waves with an adaptive threshold $Threshold_{horWav}$,

$$Threshold_{horWav} = \sum_{i=0}^{wid} \frac{HEPM_i}{wid} \tag{22}$$

Where $HEPM_i$ is the amount of horizontal edge pixels in each column of *Rois*.

Step 2. Get the x-coordinates of the left and right borders of vehicle detection region.

For a given vehicle horizontal wave, left border x_l is the first column of vehicle waves whose $HEPM_i$ satisfies Equation (23), and x_r is the last column of vehicle waves where $HEPM_i$ satisfies Equation (23).

$$HEPM_i \geq Threshold_{horWav} \tag{23}$$

Step 3. Form a vehicle detection region (VDR_n) whose width is $abs|x_r - x_l|$ and height is parameter *her* of *Rois*.

Step 4. Find the y-coordinates of the top and bottom borders of vehicle region.

Plenty of horizontal lines are contained in each VDR_n. Experimental results show that the distance between two adjacent horizontal lines belonging to the same vehicles is short. An adaptive threshold $Th\_Dis = wid\_VDR/Th\_M$ is set according to the width of VDR_n to satisfy various sizes of vehicles that are captured from UAVs at different distances and altitudes, where *wid_VDR* is the width of VDR_n and *Th_M* is a constant value obtained from numerous vehicle image statistics. In the follow-up experiment, *Th_M*is set to 5. Therefore, two adjacent lines belonging to the same vehicle area could be verified by the rules described in Algorithm 2.

Step.5 Output the x-coordinates and y-coordinates of the left-top and right-bottom of vehicles.

Algorithm 2. Vehicles top and bottom coordinates generation rules

Input: All horizontal lines $Hline_i$ from top to bottom in VDR_n.

Step 1. If $Hline_i$ is the first line in VDR_n, the top boundary of the first vehicle is obtained.

Step 2. If $Hline_i$ is the last line in VDR_n, the top boundary of the last vehicle is obtained.

Step 3. If the distance between $Hline_i$ and $Hline_{i+1}$ is smaller than *Th_Dis*, $Hline_i$ and $Hline_{i+1}$ are belonging to the same vehicle. Otherwise, $Hline_i$ is the bottom line of the jth vehicle, and $Hline_{i+1}$ is the top line of the j+1th vehicle in VDR_n.

Output: Pairs of top and bottom horizontal lines are regarded as the top and bottom borders of vehicles; therefore, the y-coordinates of the top and bottom borders of vehicle are obtained.

5. Experiments and Comparison

In order to evaluate the proposed UAV vehicle detection algorithm, the experiment platform was implemented in c using Open CV 2.4.8 library, Visual Studio 2010, and Code Composer Studio 5.5.0. The vehicle detection system is performed on video sequences with 960×540 pixels resolutions in an Intel Core i5-4590 CPU@3.30GHz PC.

(a) Detected ROIs of the 70th frame of 'Test video 1' (b) Results of Vehicle Verification Algorithm based on HEW

Figure 6. Results of vehicle verification algorithm based on horizontal edge wave

Our algorithm was evaluated using both simple and challenging conditions, and the videos were captured from a quad copter (model: dji PHANTOM 3 STANDARD) on urban road in different seasons (summer and winter). The performance evaluations of vehicle detection were based on low altitude UAV videos captured from three different scenarios with different driving conditions. The details of test videos are shown in Table 1. 'Test video 1' is a multi-vehicle detection video sequence that was captured on urban road in winter, and the resolution of the video is 960 ×540. The scene in the video is simple, and the largest vehicle amount in one frame is two. 'Test video 2' was captured on urban road in summer, and the resolution of the video is 960 ×540. The scene is complicated and includes buildings, planes, and traffic signs. The largest vehicle amount in the video is three. 'Test video 3' was captured on urban road in summer, and the resolution of the video is also 960 ×540. The scene in the video is as complicated as 'Test video 2'. The largest vehicle amount in the video is seven.

Table 1. Details of test videos

Videos	Frame amount	Vehicle Amount	Resolution	Road condition	Weather condition	Largest vehicle amount
Test video 1	500	520	960×540	urban	cloudy	2
Test video 2	216	332	960×540	urban	sunny	3
Test video 3	500	1394	960×540	urban	sunny	7

In this paper, four indicators are selected to evaluate the detection accuracy as shown in [17] including: *Detection speed*, which is in terms of processing time of each frame (ms/frame), *Correctness*, *Completeness*, and *Quality*.

$$Correctness = \frac{TP}{TP + FP} \tag{24}$$

$$Completeness = \frac{TP}{TP + FN} \tag{25}$$

$$Quality = \frac{TP}{TP + FP + FN} \tag{26}$$

Where *TP* is the number of detected vehicles, *FP* is the number of false alarms, *FN* is the number of vehicles missed, and *Quality* contains both possible detection errors (false positive and false negatives). Vehicle sizes are larger than 10×10.

To evaluate the effectiveness of our method, the FT+Mean-shift based algorithm and the UAV vehicle detection methods based on projection frame different (*PFD*) were compared. The source code of *FT+Mean-shift* was used for comparison, and it can be downloaded from their websites [29]. The general process of *PFD* is as follows: firstly, the projection match values of horizontal and vertical direction are calculated, and the offsets of horizontal and vertical direction between adjacent frames are generated. Then, the frame different method is applied to detection the motion objects. Finally, the OTSU method is used to segment the vehicle in the image. The comparison results of three algorithms are shown in Figure 7, and the comparison of the *Correctness*, *Completeness*, and *Quality* indicators for each of the three algorithms is illustrated in Table 2.

As shown in Figure 7 and Table 2, for 'Test video 1', the *PFD* method can quickly detect vehicles in images and the *Completeness* indicator reaches 100%. However, when the UAV moves faster, the shadow area of vehicles and other objects on the roadside are also detected, with the *Correctness* and *Quality* indicators both 34.2%. Although the *FT+Mean-shift* method can detect vehicles well, other objects on the road surface are also detected at the same time, resulting in many false

positives. Because our method has been improved based on *FT+Mean-shift*, adding the vehicle judgment algorithm, and excluding a large number of interfering objects, *Correctness* and *Quality* are superior to *PFD* and *FT+Mean-shift*. However, our approach results in a small number of missed tests, so *Completeness* was 96.2% lower than *PFD* and *FT+Mean-shift*. For 'Test video 2', there is a problem that *PFD* has false detection frames, and many false positives and missing detections occur. Although the *FT+Mean-shift* method can effectively detect the vehicles appearing in the images, some non-vehicle areas are also detected due to the complexity of the video scene. Because the accurate bounding box generation method based on vehicle waves excludes most of the interference area, the *Correctness* and *Quality* indexes are better than *FT+Mean-shift*.

(a) Original (b) PFDD (c) FT+Mean-shift (d) Our Method

Figure 7. Algorithms results comparison

Table 2. Algorithm indicators comparison

Scene	Metrics	PFD	FT + Mean-shift [24]	Our Method
Test video 1	Correctness (%)	34.2	27.3	85.8
	Completeness (%)	100	100	96.2
	Quality (%)	34.2	27.3	82.9
Test video 2	Correctness (%)	47.7	9.3	93.3
	Completeness (%)	69.9	100	100
	Quality (%)	39.6	9.3	93.3
Test video 3	Correctness (%)	12.1	12.9	90.4
	Completeness (%)	95.2	96.0	95.5
	Quality (%)	10.5	11.4	44.1

The CL comparison between methods is shown in Table 3. The Max, Min, and Avg are defined as the longest, shortest, and average processing time of each method. The *PFD* method is simple and easy to implement, and the CL is shown to be less than the computation load for our method. However, the detection results shown in Table 2 are much worse than our method. The mean-shift based segmentation method can segment the target position, but the algorithm processing time is too long (exceeds 30s) and cannot meet the engineering requirements.

Table 3. Computational load comparison (ms/frame)

Scene	PFD			FT + Mean-shift			Our Method		
	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
Test video 1	106	38	43	31652	30474	30864	229	201	204
Test video 2	60	54	56	36836	35208	35484	228	218	220
Test video 3	112	54	55	38761	36025	37204	235	218	221

In summary, the vehicle detection on unmanned aerial vehicle images based on saliency region detection proposed in this paper achieves a very robust performance. This represents a significant increase in the vehicle detection rate and a considerable decrease in the average processing time.

6. Conclusions

In this paper, we present a novel vehicle detection method on UAV images based on saliency region detection. There are three major contributions in this paper. First, a faster salient region detection method based on optimized frequency-turned to detect multiple vehicles in complex environments is proposed. Second, segmentation methods based on Boolean map and

OTSU are combined to determine the ROI of vehicle targets in saliency map images. It can be used to solve the problem of high time consumption of mean-shift segmentation. Finally, we propose a series of vehicle apparent features-based methods based on geometry, symmetry, and horizontal edge wave, which are used to determine vehicles and eliminate the interference of roadside objects accurately. Experimental results indicate that our method can effectively and robustly detect multiple vehicles in complicated urban environments. A comparison analysis indicates that our algorithm is more robust under challenging conditions. Despite the improvements made in our algorithm, several issues remain. These issues include CL of our method, which increases as the number of vehicles increases. Until further research has been done to improve upon these issues, the novel method presented here outperforms current methods and shows high prospects for industrial applications.

Acknowledgements

The work described in this paper was funded by the Science and Technology Development Plan of Jilin Province (No. 20170204020GX) and National Natural Science Foundation of China (No. 61602432, U1564211, and 51805203).

References

1. G. Liu, S. Liu, K. Muhammad, A. K. Sangaiah, and F. Doctor, "Object Tracking in Vary Lighting Conditions for Fog based Intelligent Surveillance of Public Spaces," *IEEE Access*, Vol. 6, pp. 29283-29296, 2018
2. C. L. Azevedo, J. L. Cardoso, M. Ben-Akiva, J. P. Costeira, and M. Marques, "Automatic Vehicle Trajectory Extraction by Aerial Remote Sensing," *Procedia - Social and Behavior Sciences*, Vol. 111, pp. 849-858, Febuary 2014
3. A. C. Shastry and R. A. Schowengerdt, "Airborne Video Registration and Traffic-Flow Parameter Estimation," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6, No. 4, pp. 391-405, December 2005
4. Y. Wu, H. Sun, and P. Liu, "A Novel Fast Detection Method of Infrared LSS-Target in Complex Urban Background," *International Journal of Wavelets, Multiresolution and Information Processing*, Vol. 16, No. 1, pp. 1850008, January 2018
5. H. Yalcin, M. Hebert, R. Collins, and M. J. Black, "A Flow-based Approach to Vehicle Detection and Background Mosaicking in Airborne Video," in *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (*CVPR'05*), Vol. 2, pp. 1202-1202, 2005
6. T. S. C. Tan, "Colour Texture Analysis using Colour Histogram," *IEE Proceedings - Vision, Image and Signal Processing*, Vol. 141, No. 6, pp. 403, 1994
7. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346-359, June 2008
8. S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *Proceedings of 2011 International Conference on Computer Vision*, pp. 2548-2555, 2011
9. A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast Retina Keypoint," in *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 510-517, 2012
10. S. Liu, X. Cheng, W. Fu, Y. Zhou, and Q. Li, "Numeric Characteristics of Generalized M-Set with its Asymptote," *Applied Mathematics and Computation*, Vol. 243, pp. 767-774, September 2014
11. X. Cao, C. Wu, J. Lan, P. Yan, and X. Li, "Vehicle Detection and Motion Analysis in Low-Altitude Airborne Video under Urban Environment," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 21, No. 10, pp. 1522-1533, October 2011
12. J. Leitloff, D. Rosenbaum, F. Kurz, O. Meynberg, and P. Reinartz, "An Operational System for Estimating Road Traffic Information from Aerial Images," *Remote Sensing*, Vol. 6, No. 11, pp. 11315-11341, November 2014
13. X. Cao, C. Wu, P. Yan, and X. Li, "Linear SVM Classification using Boosting HOG Features for Vehicle Detection in Low-Altitude Airborne Videos," in *Proceedings of 2011 18th IEEE International Conference on Image Processing*, pp. 2421-2424, 2011
14. S. Tuermer, F. Kurz, P. Reinartz, and U. Stilla, "Airborne Vehicle Detection in Dense Urban Areas using HoG Features and Disparity Maps," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 6, No. 6, pp. 2327-2337, December 2013
15. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, pp. 1627-1645, September 2010
16. Y. Xu, G. Yu, Y. Wang, and X. Wu, "Vehicle Detection and Tracking from Airborne Images," in *Proceedings of CICTP 2015*, pp. 641-649, 2015
17. Y. Xu, G. Yu, X. Wu, Y. Wang, and Y. Ma, "An Enhanced Viola-Jones Vehicle Detection Method from Unmanned Aerial Vehicles Imagery," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 7, pp. 1845-1856, July 2017
18. W. Li, P. Liu, Y. Wang, H. Ni, C. Wen, and J. Fan, "On-Board Robust Vehicle Detection using Knowledge-based Features and Motion Trajectory," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 8, No. 2, pp. 201-212, February 2015
19. S. Liu, Z. Zhang, L. Qi, and M. Ma, "A Fractal Image Encoding Method based on Statistical Loss Used in Agricultural Image Compression," *Multimedia Tools and Applications*, Vol. 75, No. 23, pp. 15525-15536, December 2016
20. R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient Region Detection and Segmentation," in *Proceedings of International Conference on Computer Vision Systems*, pp. 66-75, Berlin, Heidelberg, 2008
21. M. M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. M. Hu, "Global Contrast based Salient Region Detection," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 2015
22. M.-M. Cheng, G. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global Contrast based Salient Region Detection," in *Proceedings of CVPR 2011*, pp. 409-416, 2011
23. Y. Zhai and M. Shah, "Visual Attention Detection in Video Sequences using Spatiotemporal Cues," in *Proceedings of the 14th Annual ACM International Conference on Multimedia*, pp. 815, 2006
24. N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga, "Saliency Estimation using a Non-Parametric Low-Level Vision Model," in *Proceedings of CVPR 2011*, pp. 433-440, 2011
25. L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "SUN: A Bayesian Framework for Saliency using Natural Statistics," *Journal of Vision*, Vol. 8, No. 7, pp. 32, December 2008
26. D. Walther, "Interactions of Visual Attention and Object Recognition: Computational Modeling, Algorithms, and Psychophysics," California Institute of Technology, 2006
27. J. Lou, W. Zhu, H. Wang, and M. Ren, "Small Target Detection Combining Regional Stability and Saliency in a Color Image," *Multimedia Tools and Applications*, Vol. 76, No. 13, pp. 14781-14798, July 2017
28. R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-Tuned Salient Region Detection," in *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1597-1604, 2009
29. R. Achanta, "Saliency_Map_Comparison," 2009, (http://ivrlwww.epfl.ch/supplementary_material/RK_CVPR09/)
30. N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, pp. 62-66, January 1979
31. X. Hou and L. Zhang, "Saliency Detection: A Spectral Residual Approach," in *Proceedings of 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007
32. W. Li, P. Liu, Y. Wang, and H. Ni, "Multifeature Fusion Vehicle Detection Algorithm based on Choquet Integral," *J. Appl. Math.*, Vol. 2014, pp. 1-11, 2014