American Journal of Applied Sciences

Original Research Paper

Vehicle Detection in Aerial Traffic Monitoring

Dmitry Sincha, Mikhail Chervonenkis and Pavel Skribtsov

PAWLIN Technologies Ltd, Dubna, Russia

Article history
Received: 23-09-2015
Revised: 29-12-2015
Accepted: 04-01-2016

Corresponding Author:
Pavel Skribtsov,
PAWLIN Technologies Ltd,
Dubna, Russia
Email: info@pawlin.ru

Abstract: This work describes a cascade detection of vehicles in Unmanned Aerial Vehicle (UAV) images and videos. There are some new approaches used in the detection. In particular, the Region of Interest (ROI) search is not only based on GIS and navigation data, but also employs visual method based on rapid image segmentation and road detection. The work also suggests doing ROI segmentation by the superpixel technique and trainable four-level cascade detector that uses artificial neural networks as classifiers. Characteristics of the being analyzed regions (combined superpixels) are based on geometric and texture features, as well as on deep features extracted from the image patches by nonlinear auto encoders. To improve the detection quality of the moving vehicles a separate stage of the detector based on optical flow analysis was introduced. Proposed detection algorithm was benchmarked on the real UAV videos and showed the sufficiently high accuracy. Performance of the algorithm allows supposing the on-board usage.

Keywords: Unmanned Aerial Vehicles (UAV), Vehicle Detection, Road Detection, Superpixel, Deep Learning

Introduction

The problem of vehicle detection on aerial photos and videos has become more important lately because of the spread of UAV. The vehicle detection can be used mostly in the automation of monitoring of traffic and large parking lots. Works about the automatic vehicle recognition have long been published, for example (Coifman, 2006).

The work by Kim and Chervonenkis (2015) describes the importance of the on-board recognition of both moving and stationary vehicles for automatic detection and classification of traffic situations. In this study a cascading approach to the vehicle recognition has been defined. The works (Kim and Chervonenkis, 2015; Abramov *et al*., 2015) have suggested using the image segmentation into superpixels followed by their association in regions, this approach is also used in proposed algorithm herein. These works were inspired by (Choi and Yang, 2009) paper which applies mean shift segmentation in the Luv color space in order to extract blobs (superpixels) of the image. Subsequently, the symmetry of the resulting blobs is examined by a filter based on complex valued Gabor functions. Additionally, the information of the shape is used. The shape of each blob is calculated by measuring the distance and orientation between the center of the blob and its surrounding edges. The authors point out that often more than one blob is detected for the same car due to intensity differences from the front and rear windshields. So blobs (superpixels) clustering procedure is needed and was introduced in aforementioned works. Description of the object area as a region allows using features of the shape of the object for its classification.

In a number of works it has been proposed to use texture features to detect vehicles: Kembhavi *et al.* (2011) employ Histogram of Oriented Gradients (HOG) features; Nguyen *et al.* (2006; Grabner, 2008; Mauthner *et al*., 2010) use Local Binary Patterns (LBP) features and HOG features; Gleason *et al.* (2011) deal with HOG features and Histogram of Gabor coefficients features. A thorough analysis of these works revealed the HOG features having the main impact on vehicle quality detection. These features are used in two stages of our cascade.

The last stages of the developed cascade deal with the features based on the movement of the object and the features built on the basis of the analysis of image fragments by means of nonlinear autoencoders.

The mentioned work by (Kim and Chervonenkis, 2015) describes in detail the methodology of the use of UAV for road traffic monitoring and proposes an approach based on the recognition and classification of severity traffic situation. In this study we described the UAV automatic control. To recognize and classify

Science Publications

amicable situations proposed to use the features of the recognized and tracked vehicles, in particular, such as the histogram of the speed distribution of the vehicles. But description of vehicle detector is not described in detail.

The general objective of our study is to develop a new multi-layer cascade vehicle detector, capable of operating on-board with sufficient speed. The novelty of the approach lies in the recognition of the area of the road for the allocation of ROI, in the selection of a set of specific geometry, texture, deep and motion features for cascade detector. Among the proposed geometric features-concentration ellipse dimensions and ellipse eccentricity, edge density, the use of which has become possible due to the recognition of objects as unified superpixels. Application of deep autoencoders features and the method of preparation of learning sample, based on estimates of the accuracy of matching the boundaries of the region and vehicle are also new to such problems.

Methodology

The functional block diagram of the algorithm of vehicle detection on the image is shown in Fig. 1.

The algorithm input receives a pair of consecutive RGB-images from an on-board UAV camera (still pictures or shots from a video flow) and the vehicle search is carried out on the earliest one (the operational one). We consider the case of a mechanically stabilized on-board camera performing a nadir shooting.

Image pre-processing scales the operational image with the following formation of a binary Region(s) of Interest (ROI) mask. The scale coefficient is calculated by the Navigation data-based Processing unit on the basis of the UAV flight altitude at the time of the operational image shooting. Scaling allows reducing the impact of the flight altitude on vehicle detection quality, as well as improving the performance of the entire system. If there is a digital terrain map in the on-board system, the Navigation data-based Processing unit generates an inclined rectangle for the operational image. This rectangle defines the position of the road with roadside and other surrounding area that can be reached by a vehicle in case of an accident. In practice, usually one rectangle is formed. However, in some cases (a sharp bend in the road, crossroad, forked roads, etc.) the unit defines the region of interest by means of several slightly intersecting rectangles. When the on-board system does not contain a sufficiently detailed and accurate digital map of the area, or in the case of low accuracy of determination of the UAV location and orientation, the ROI search is performed by a Road Detection unit.

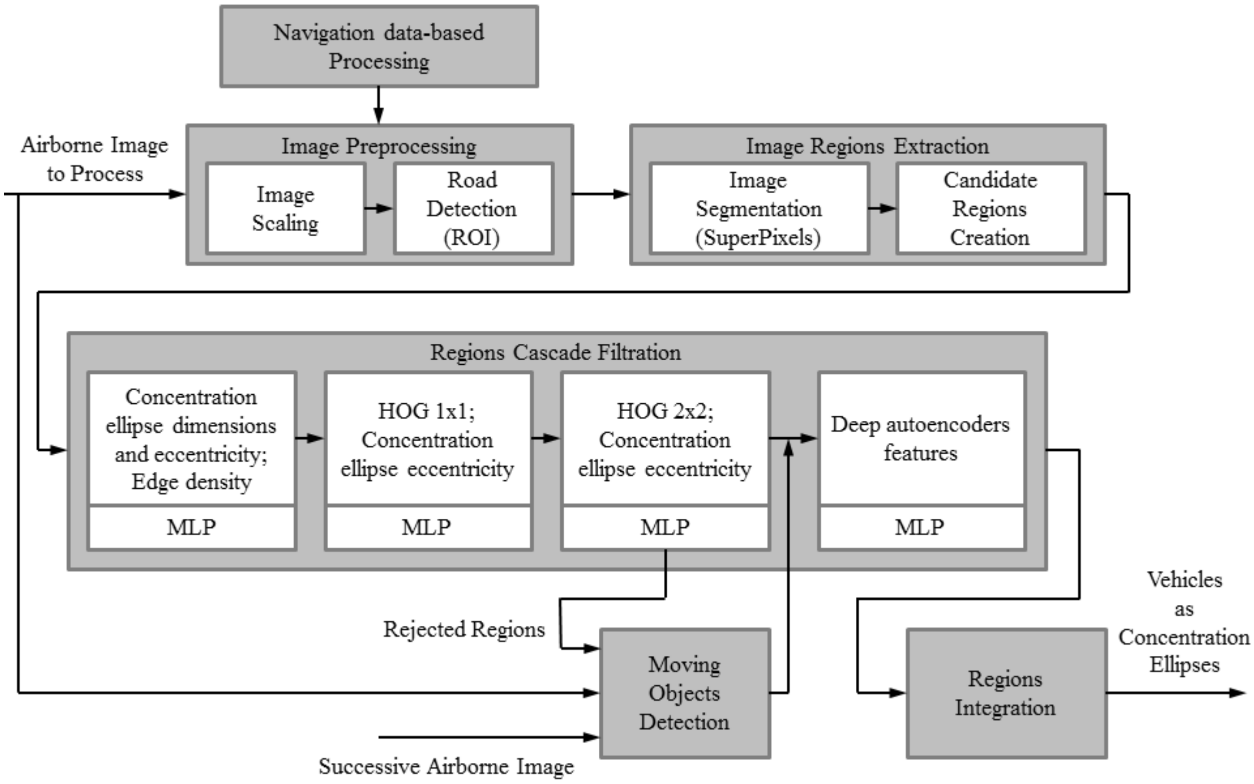
Navigation data-based Processing
Airborne Image to Process
Image Preprocessing
Image Scaling
Road Detection (ROI)
Image Regions Extraction
Image Segmentation (SuperPixels)
Candidate Regions Creation
Regions Cascade Filtration
Concentration ellipse dimensions and eccentricity; Edge density
HOG 1x1; Concentration ellipse eccentricity
HOG 2x2; Concentration ellipse eccentricity
Deep autoencoders features
MLP
MLP
MLP
MLP
Rejected Regions
Moving Objects Detection
Successive Airborne Image
Regions Integration
Vehicles as Concentration Ellipses

Fig. 1. Functional block diagram of the developed vehicle detection algorithm

Unit of Visual Automatic ROI Determination

Visual method of automatic ROI determination consists in determination in an image of an area (or mask) of unspecified form which contains the whole or almost the whole road. The algorithm is as follows:

1) First of all, the quick initial image segmentation is carried out. The individual segments are related and relatively homogeneous regions. In this case the color image is analyzed. Color space is initially divided into a limited number of fixed clusters. In the simplest case, this can be a small (for example, 8) number of brightness gradations. In the case of a more complex performance, all colors of pixels are taken into account. A region of pixel connectivity belonging to the same color cluster is lumped into one segment. Simultaneously with the segment construction, its moments of first and second order are quickly calculated. On the basis of central moments of the second order, the parameters of segment dispersion ellipse are calculated: Axes size, elongation (eccentricity) and the orientation of the main axis. (Hereinafter the major axis of the ellipse is referred to as the segment axis). The moments of the first order are considered as coordinates of the midpoint of a segment.

2) Next step is the filtration of received segments. First of all it is necessary to exclude bright segments. Then segments that are small, not enough elongated and very large are excluded from the list. Filtration threshold is selected.

3) Several segments are put together. For this purpose, all possible pairs of segments are considered. Segments are joined together only under certain conditions. In particular, the distance between midpoints of segments shall be less than the axis of each segment multiplied by a parameter, the orientation of segments must vary little (less than a parameter, in our case it is 5-10°). There may also be restrictions on the proximity of color of joined segments. For joined segments moments are calculated on the basis of moments of each joined segment.

4) The segments obtained as a result of joining, are joined to one another in a similar manner.

All received segments are checked for similarity to the road. The axis of the segment shall be close to both the two opposite sides of the original image. All segments satisfying such conditions are joined in one set-this is the desired ROI and a mask.

Image Regions Extraction performs segmentation of the operational image (with due regard to the ROI mask) followed by a region creation using the resulting segments. The final regions are considered as potential vehicles.

In order to select a suitable segmentation algorithm, the following superpixel extraction algorithms were compared: Felzenszwalb-Huttenlocher Segmentation (FHS) (Felzenszwalb and Huttenlocher, 2004), SEEDS (Van den Bergh *et al.*, 2012), SEEDS Revised (Stutz, 2014; 2015), SLIC (Radhakrishna *et al.*, 2012), Model Based Clustering (MBC) (Zhong and Ghosh, 2003), Quick shift (Vedaldi and Soatto, 2008). High average image processing time makes MBC and Quick shift algorithms unsuitable for on-board use. SLIC algorithm is unsuitable due to a poor quality of segmentation (for a detailed comparison of these three algorithms, see Abramov *et al.*, 2015). Comparison of FHS, SEEDS and SEEDS Revised algorithms has been conducted on operational images without regard to the ROI mask as per the following characteristics (Table 1): (a) Average image processing time; (b) Average number of image superpixels; (c) Average number of superpixels into which the vehicle is divided; (d) Average vehicle segmentation performance value-Undersegmentation Error and Boundary Recall.

The (a)-(c) characteristics affect the performance of the entire vehicle detection system. The segmentation quality indicators used are given in (Neubert and Protzel, 2007). The Undersegmentation Error (UE) indicator (Formula 1) shows how well a set of superpixels covering the vehicle follows its shape:

$$UE = \frac{1}{N} \sum_{S \in Vehicle} \min(P_{in}, P_{out}) \quad (1)$$

Where:

P_{in} = The number of vehicle covering pixels of the *S* superpixel
P_{out} = The number of pixels of the S superpixel that are outside the boundaries of the vehicle
N = Vehicle area (number of pixels)

The *Boundary Recall* indicator shows the proximity of the borders of vehicle covering superpixels to the borders of this vehicle. The indicator is calculated as a percent of vehicle boundary pixels having in a predetermined radius around themselves the boundary pixels of vehicle covering superpixels. Images illustrating segmentation quality indicators used are shown in Fig. 2.

(Left) The red rectangle indicates the vehicle. *B*, *C*, *E* and *F* superpixels cover the vehicle. The area of the green part of each such superpixel-P_{in} and yellow-P_{out} (Formula 1). (Right) Rectangles show the boundary pixels: The black ones belong to the vehicle, the blue ones belong to the vehicle covering superpixel. The upper boundary vehicle pixel does not have in its *d* radius of boundary superpixel covering pixels; the lower boundary vehicle pixel has such pixels

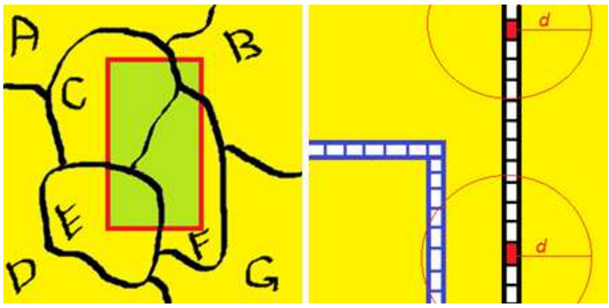

Fig. 2. Explanatory images for Undersegmentation Error (left) and Boundary Recall (right) indicators

Table 1. Comparison of image segmentation algorithms

	FHS	SEEDS	SEEDS revised
Image processing time, ms	249	96	369
Boundary Recall, [0; 1]	0,52	0,45	0,5
Undersegmentation Error, [0; 1]	0,34	0,57	0,2
Number of superpixels per image	831	1198	1200
Number of superpixels per vehicle	3,5	4,5	4,3

SEEDS Revised algorithm is characterized by the maximum operational image processing time (Table 1). In contrast, the SEEDS algorithm is the fastest one, but it breaks the vehicle into the largest number of superpixels that leads to the creation of the largest number of regions. The most balanced algorithm is FHS, who has been selected as the operational image segmentation algorithm.

The first step of the FHS algorithm introduced in Felzenszwalb and Huttenlocher (2004) paper is weighted graph building. Each edge of the graph represents the joint of the two adjacent image pixels. The weight of the edge is dissimilarity measure between pixels. The usual practice for weight calculation is Euclidian distance between pixels' RGB values. In this study we use weighted Euclidian distance in Lab color space. In the next step of the algorithm the edges of the graph are sorted in weight ascending order. Running through the sorted list of edges the superpixels (represented by the disjoint trees) are created by means of joining similar image pixels. The size of the superpixels is adjusted by the pixels' dissimilarity threshold.

When using the FHS algorithm the vehicle can usually be well approximated by one to four superpixels (Table 1). Therefore, in addition to the superpixel analysis, it is necessary to conduct the analysis of complexes of neighboring superpixels: Pairs, "threes" and "fours". The test of the developed vehicle detection algorithm, however, has showed that when using "fours", the vehicle detection quality indicators are improving slightly, but the performance is reduced significantly. Therefore, all possible superpixels shall be considered as image regions, including pairs of adjacent superpixels and combinations of three superpixels, in which at least one is adjacent to the other two. As a result of the scaling of the operational image (Fig. 1) it is possible to establish rough thresholds above and below the size of the region and not to create areas which are clearly not a vehicle.

In addition to information about the superpixel components, each region is described by a concentration ellipse. C is a matrix of second central moments calculated as per all N pixels in the region $X_i^T = (x_i; y_i)^T$:

$$C = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \overline{X})(X_i - \overline{X})^T = \begin{pmatrix} c_{11} & c_{12} \\ c_{12} & c_{22} \end{pmatrix} \qquad (2)$$

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \qquad (3)$$

The midpoint of the concentration ellipse is defined by the formula 3. Large (a) and small (b) axis of the ellipse and its orientation-the angle Θ between the major axis and the positive direction of the OX axis in the working image coordinate system are calculated as follows:

$$a = \sqrt{0.5\cdot(c_{11} + c_{22} + \sqrt{D})} \qquad (4)$$

$$b = \sqrt{0.5\cdot(c_{11} + c_{22} - \sqrt{D})} \qquad (5)$$

$$\Theta = \arctan\left(\frac{c_{12}}{a^2 - c_{11}}\right) \qquad (6)$$

$$D = (c_{11} - c_{22})^2 + 4c_{12}^2 \qquad (7)$$

We use the concentration ellipse with semi-axes of the size twice bigger than the size of semi-axes calculated by formulas 4 and 5. The advantage of use of the concentration ellipse as a compressed representation of the region (above an inclined rectangle, a convex hull, etc.) is a greater resistance of its parameters to segmentation errors. An example of the operational image (with an applied ROI mask) and some of its regions is shown in Fig. 3.

Regions Cascade Filtration provides a binary classification of the regions represented by concentration ellipses by the following classes: "Vehicle" and "Everything else", followed by filtration of regions referred to the second class. Classification is carried out according to the cascade principle: Region features the calculation of which is more time-taking are calculated on the later stages of the cascade. As a classifier at each stage of the cascade a Multilayer Perceptron (MLP) is used.

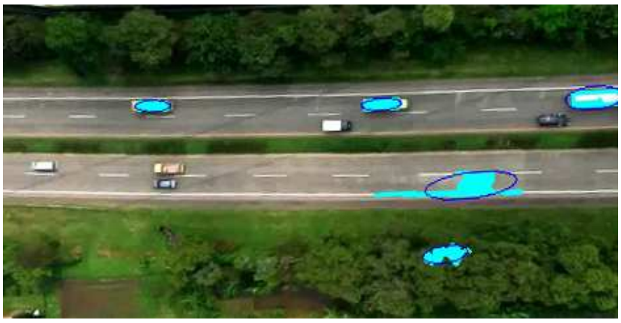

Fig. 3. Operational image (region of interest only) and some of its regions (blue areas). Concentration ellipse for each region is shown in blue

At the first stage of the cascade, regions are described using the following features: The length of semi-axes (a and b; $a \geq b$), the eccentricity of the concentration ellipse:

$$e = \sqrt{1 - \frac{b^2}{a^2}} \tag{8}$$

And edge density:

$$\rho_E = \frac{N_E^2}{W \cdot H} \tag{9}$$

(here N_E-the number of edge pixels fallen into the window of $W \times H$ size). In order to calculate the last feature (ρ_E) the operational RGB-picture is brought to a single-channel half-toned one from which the edge mask is extracted using the Canny algorithm. Two thresholds used by the Canny algorithm, are selected automatically during the training of the classifier of this stage of the cascade. After that an edge mask integral image is built, which allows calculating the N_E value in 3 arithmetic operations (1 addition and 2 subtractions). The window for the region is obtained by an extension of a minimum direct rectangle delineating the 50% of concentration ellipse on each side. The configuration of the neural network classifier of the cascade stage is 4:14:1.

At the second and third stages of the cascade, in addition to the eccentricity of the concentration ellipse for regions, a HOG-descriptor (histogram of gradient orientations) is calculated. The original version of the descriptor was developed by (Dalal and Triggs, 2005; Dalal, 2006) for the purpose of solving the problem of pedestrian detection in static images. Currently the descriptor and its modifications are successfully used in many algorithms for detecting different objects. In particular, the vast majority of modern algorithms for vehicle detection in static images, not using an explicit model of the vehicle, comprise the calculation of HOG-descriptor (Turmer, 2014).

In our implementation which differs from the original one, the HOG-descriptor is calculated as follows. It is necessary to form for each region a region-centered square window of size (where a and b-the length of the semi-axes of the concentration ellipse). For each pixel of the window the length $M_G(x, y)$ and direction $\Theta_G(x, y)$ of the gradient vector are calculated as follows:

$$W = H = 2 \cdot (a + b) \tag{10}$$

$$M_G(x,y) = \max\{M_{G(R)}(x,y), M_{G(G)}(x,y), M_{G(B)}(x,y)\} \tag{11}$$

$$M_{G(R/G/B)}(x,y) = \sqrt{\left(\frac{\partial I_{R/G/B}(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I_{R/G/B}(x,y)}{\partial y}\right)^2} \tag{12}$$

$$\Theta_G(x,y) = \arctan\left(\frac{\partial I(x,y)}{\partial y} \Big/ \frac{\partial I(x,y)}{\partial x}\right) \tag{13}$$

The direction of the gradient vector is calculated in the same color channel (R/G/B) as its length and is brought to the $\left[-\frac{\pi}{2}; \frac{\pi}{2}\right)$ interval. Intensity derivatives are calculated using the Sobel operator with a 3×3 core. After that, a histogram of gradient directions (orientations) consisting of 9 bins is calculated in the windows. The weight of each bin is the sum of lengths of all vectors of the gradient, fallen in the bin. The resulting h histogram is normalized-it is divided by its L2-norm:

$$h_n = h \Big/ \sqrt{\|h\|_2^2 + \xi^2} \tag{14}$$

(here ξ-a small constant, in our implementation $\xi = 0.01$). If the examined region is well approaching the vehicle, the gradient vector of the dominant direction will be orthogonal towards the longitudinal axis of the vehicle. Consequently, in order to increase the descriptor resistance to rotation, the h_n normalized histogram components undergo cyclic shift so that the bin with the maximum weight is always in the first position. The resulting histogram is a 1×1 HOG-descriptor (Fig. 1). The configuration of the neural network classifier of the second stage of the cascade is 10:23:1.

2×2 HOG-descriptor (Fig. 1) is calculated in a similar manner. For each window a region is formed as described above for construction of the 1×1 HOG-descriptor. The window is divided into four overlapping by 25% square cells of equal size (Fig. 4), in each of which the h_n gradient orientation normalized histogram consisting of 8 bins is calculated.

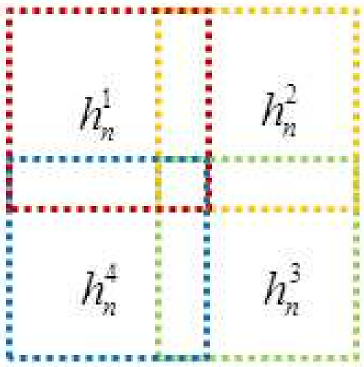

Fig. 4. View of the window used in order to calculate the 2×2 HOG-descriptor

The descriptor is obtained by concatenating h_n histograms taken in "clockwise" order. Due to the low resistance of the descriptor to rotation, the training of the classifier of the third stage of cascade is conducted with "reproduction" of positive samples (of relevant vehicles). Each positive sample creates three more samples that model the rotation of the vehicle by ±90 and 180° with respect to its current orientation. New samples are obtained by cyclic permutation of the descriptor histograms with simultaneous cyclic shift of their components. The configuration of the neural network classifier of the third stage of the cascade is 33:29:1.

In both analyzed HOG-descriptors non-normalized *h* gradient orientation histogram is calculated by means of integral images. For this purpose, it is necessary to build a block of a *bins* (*h*) size (where *bins* (*h*)-the number of bins in the *h* histogram) of gradient maps. Each such a card is a single-channel image of a (*W*, *H*) size equal to the size of the operational image after applying the ROI mask. Each pixel of gradient map contains a record about the length of the gradient vector at this point of the operational image (see formula 8) or a zero. Each gradient map corresponds to a bean in the *h* histogram, i.e., covers a range of directions of the gradient vector of 20° width for 1×1 HOG and 22.5° for 2×2 HOG. Then, the integral images of all gradient maps are calculated. Thus, non-normalized *h* gradient orientation histogram in the window is calculated by means of 3.*bins* (*h*) arithmetic operations.

Analyzed HOG-descriptors possess a high discriminating ability. It is interesting to note that the *N*×*N* modification HOG-descriptor (where *N*-is a number of units by which the window is divided as per width/height) which suggests itself, where *N*>*2*-leads to a more complex algorithm of "reproduction" of positive samples and according to our analysis, it has a low discriminating ability. Our study has been also confirmed by (Gleason *et al*., 2011).

Method of Traffic Areas Determination

The method is based on a comparison of neighboring shots. For adjacent (*X* and *Y*) shots an optimal T rigid transform that allows combining images is built. This transform describes well the change in the image due to the motion of the camera (and UAV) on the assumption of its optical stabilization. At this stage, a single-channel image is used to speed up the calculation. The transform building is made very quickly and using an exactly known method based on the construction of the Lucas-Canade optical flow on the image pyramid. If *T(X)*=is an *X* shot influenced by the *T* transform, then, on the:

$$Z = abs\left(Y - T\left(X\right)\right) \tag{15}$$

Sample (here abs is a capturing pixel by pixel of the absolute value) most of the points without motion have a low intensity. After smoothing of Z and threshold filtering, we get the image on which the majority of non-zero intensity pixels correspond to points of a true motion of objects in the image.

After binarization and construction of an integral image, it becomes easy to use a simple way of calculating the number in which there is the movement for any straight-oriented rectangle. The approach described above can be quickly implemented.

The Fourth Stage

For each region we create a rectangular image in which the vehicle is positioned vertically. Image size = size of the inclined rectangle circumscribing about a concentration ellipse + 50% on each side. Image is launched in two deep autoencoders. Features of the region at this stage of the cascade are obtained by concatenating of the encoder outputs. The configuration of the first autoencoder is: 1088:100:10:100:1088. The configuration of the second autoencoder is: 374:30:5:30:374. Total region is described by 15 features. The configuration of the classifier at this stage is: 15:14:1.

In order to train these two autoencoders, the initial set of positives (Fig. 5) (for training purpose only positives are used) is clustered into two clusters by size. The centers of clusters determine the size of pictures that autoencoders are working with. Training (but not validation) positives are reproduced by rotation by 180°. The training of autoencoder is performed by minimizing the sum of reconstruction error squares. Training images are brought to the gray; their average is deducted from them and divided by the norm. In this case, the optimization criterion becomes the minimization of the sum of correlation coefficients. Multilayer autoencoder is trained as a sequence of single-layer ones, followed by gluing and fine tuning.

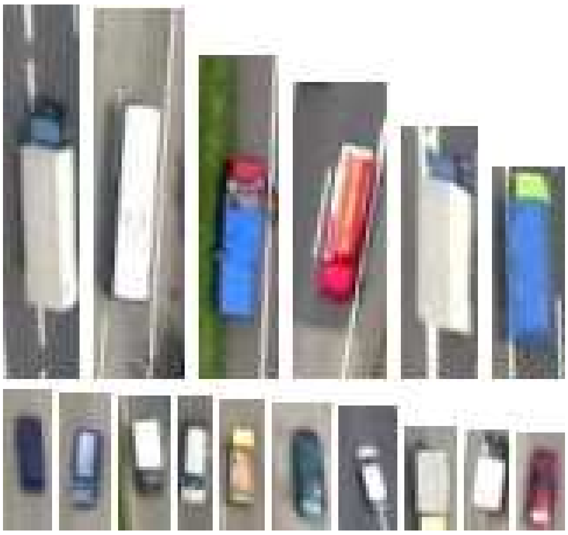

Fig. 5. Samples of training images for autoencoders

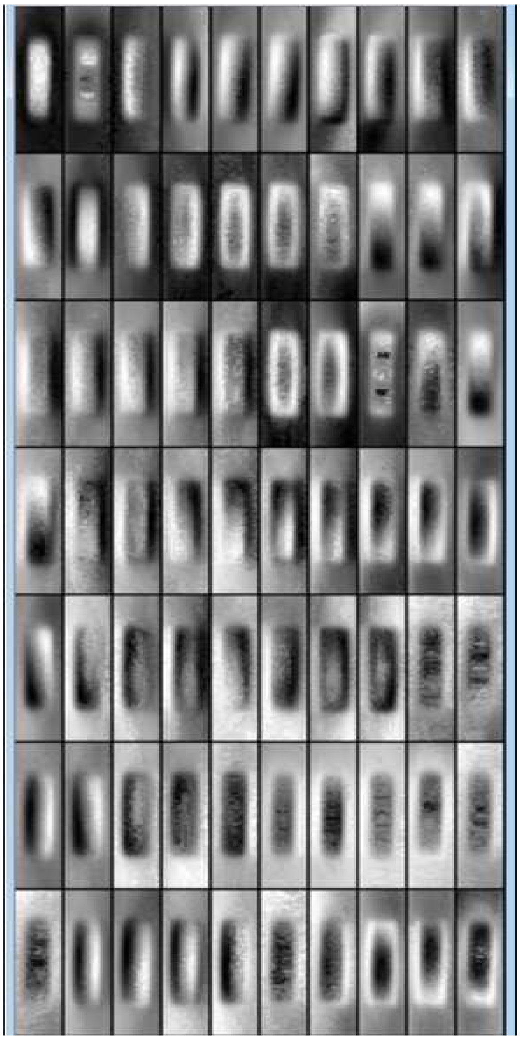

Fig. 6. Visualization of hidden neurons of the first layer

Table 2. Confusion matrix (case: Moving objects detection unit is off)

	Actual negatives	Actual positives (Vehicles)
Predicted negatives	0.992	0.136
Predicted positives	0.008	0.864

Table 3. Confusion matrix (case: Moving objects detection unit is on)

	Actual negatives	Actual positives (Vehicles)
Predicted negatives	0.955	0.011
Predicted positives	0.045	0.989

The first hidden layer is initialized by a two-dimensional cosine transform. Following hidden layers are PCA. When training the first monolayer autoencoder KL regularization and weight decay regularization are used (Ng, 2011). When training the next monolayer autoencoders only KL is used. The fine tuning is carried out without regularization.

Samples of training images for the first and second autoencoders and visualization of hidden neurons of the first layer are shown in Fig. 6 (The configuration of the autoencoder is 1088:100:10:100:1088).

Results

Our experiments on the real marked UAV videos result to the following detection quality rates Table 2 and 3.

The performance of the developed vehicle detection system in case of working with a full 640×480 shot without ROI selection was 5 fps (when using mean ×86 single core processor and the FHS method of segmentation). Visual ROI detection technique boosts the performance up to 19 fps. It’s possible to increase the performance at the cost of detection quality-we should switch to the SEEDS segmentation method.

Discussion

The peculiarity of the proposed detection method is as follows:

1) New effective and rapid methods for ROI selection which allow multiply reducing the search area have been proposed. ROI selection can significantly reduce the time of further processing and the number of false responses.

2) It has been proposed to use the methods of superpixels selection for segmentation of high-precision ROI segmentation. These methods are not so rapid, but because of ROI detection, selection of the appropriate scale and optimization of segmentation algorithms it is possible to get acceptable performance.

3) A set of superpixels and combinations of neighboring superpixels-as regions-candidates for the presence of images of an individual vehicle are built. Restrictions on these regions make it possible to restrict the size of the resulting set of regions.

4) For the vehicle detection in formed regions, a cascade of trainable classifiers is used. The first level of the classifier is very rapid; the latter one is less rapid. All levels are using a MLP technique.

5) The first level of the cascade is based on the analysis of integral characteristics of the shape of the region (moments). This simple classifier allows significantly reducing the initial selection.

6) The second and third levels of the cascade are based on analysis of (HoG) texture of image elements in the region. These levels also allow significantly reducing the selection.

7) The last level of the cascade is built on the feature selection by the method of construction of a regularized neural network image autoencoder and the encompassing inclined rectangular area of each region is brought to a single rectangular form by a linear transformation.

8) Moving areas in the image are detected by a separate fast algorithm, which is based on a comparison of neighboring shots. The results of this analysis are used at the final classification levels in order to significantly improve the accuracy of classification of moving vehicles.

It is possible to conduct further studies in the following areas:

1) On the obtained ROI (which usually contains the entire road visible in the shot) quite small areas may be selected. Further object search will be made only in those areas. Algorithm of selection of these areas is similar to the algorithm of ROI selection and will also be fast. This approach will allow speeding up the algorithm by several times in general and reducing false responses.

2) Color characteristics can be used in the detection. Now they are used only for segmentation. This can significantly increase the accuracy of detectors.

3) Autoencoder can also use multi-channel (color) image.

4) For images of different sizes different autoencoders can be built.

5) The algorithm can be supplemented by an object type classifier. For example, light-duty vehicles and load carrier vehicles.

6) The algorithm can be used to detect other objects not vehicles, for example, animals.

7) In the algorithm, various types of classifiers can be used, not necessarily MLP.

8) In the algorithm, various methods of segmentation can be used, not necessarily FHS.

Conclusion

The method allows detecting motionless vehicles with a good accuracy and moving vehicles with an even greater accuracy.

The resulting detection was tested on actual experimental data and showed the sufficiently high accuracy.

The speed of the algorithm allows supposing the on-board use. The developed method can be used for traffic monitoring, evaluation of parking occupation and a number of related tasks.

Acknowledgement

The authors gratefully acknowledge support of the Ministry of Education and Science of the Russian Federation provided for this research.

Funding Information

The research was financed by the Ministry of Education and Science of the Russian Federation under the Grant Contract dated 8 September, 2014, (grant number 14.576.21.0051, contract unique identifier RFMEFI57614X0051), the grant was assigned for the research into “Development of intelligent algorithms of traffic situations detection and identification for the on-board systems of the unmanned aerial vehicles performing automatic traffic patrolling by means of GLONASS”.

Author’s Contributions

Dmitry Sincha: Wrote paragraphs Methodology and Results and Discussion and developed the method of Regions Cascade Filtration.

Mikhail Chervonenkis: Wrote paragraphs Introduction, Conclusion and Methodology and developed the method of traffic areas determination.

Pavel Skribtsov: Designed the research plan and organized the study and also developed the unit of visual automatic ROI determination.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Abramov, K.V., P.V. Skribtsov and P.A. Kazantsev, 2015. Image segmentation method selection for vehicle detection using unmanned aerial vehicle. Modern Applied Sci., 9: 295-303. DOI: 10.5539/mas.v9n5p295

Choi, J.Y. and Y.K. Yang, 2009. Vehicle Detection from Aerial Images using Local Shape Information. In: Advances in Image and Video Technology, Wada, T., F. Huang and S. Lin (Eds.), Springer Science and Business Media, ISBN-10: 3540929568, pp: 227-236.

Coifman, B., 2006. Roadway traffic monitoring from an unmanned aerial vehicle. IEE Proc. Intellig. Trans. Syst., 153: 11-20. DOI: 10.1049/ip-its:20055014

Dalal, N., 2006. Finding People in Images and Videos. PhD Thesis, Institut National Polytechnique de Grenoble.

Dalal, N. and B. Triggs, 2005. Histograms of oriented gradients for human detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun. 25-25, IEEE Xplore Press, San Diego, CA, USA, pp: 886-893. DOI: 10.1109/CVPR.2005.177

Felzenszwalb, P.F. and D.P. Huttenlocher, 2004. Efficient graph-based image segmentation. Int. J. Comput. Vision, 59: 167-181. DOI: 10.1023/B:VISI.0000022288.19776.77

Gleason, J., A.V. Nefian, X. Bouyssounousse, T. Fong and G. Bebis, 2011. Vehicle detection from aerial imagery. Proceedings of the IEEE International Conference on Robotics and Automation, May 9-13, Shanghai, pp: 2065-2070. DOI: 10.1109/ICRA.2011.5979853

Grabner, H., 2008. On-line Boosting and Vision. PhD Thesis, TU Graz.

Kembhavi, A., D. Harwood and L. Davis, 2011. Vehicle detection using partial least squares. IEEE Trans. Patt. Analysis Machine Intellig., 33: 1250-1265. DOI: 10.1109/TPAMI.2010.182

Kim, N.V. and M.A. Chervonenkis, 2015. Situation control of unmanned aerial vehicles for road traffic monitoring. Modern Applied Sci., 9: 1-13. DOI: 10.5539/mas.v9n5p1

Mauthner, T., S. Kluckner, P. Roth and H. Bischof, 2010. Efficient object detection using orthogonal NMF descriptor hierarchies. Proceedings of the 32nd DAGM Conference on Pattern Recognition, Sep. 22-24, Darmstadt, Germany, pp: 212-221. DOI: 10.1007/978-3-642-15986-2_22

Neubert, P. and P. Protzel, 2007. Superpixels benchmark and comparison. Proceedings of Forum Bildverarbeitung, Scientific Publishing, (BSP' 07), Karlsruhe, pp: 1-12.

Ng, A.Y., 2011. Sparse autoencoder. Stanford Univ., CS294A Lecture notes.

Nguyen, T.T., H. Grabner, B. Gruber and H. Bischof, 2006. On-line boosting for car detection from aerial images. Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future, Mar. 5-9, IEEE Xplore Press, Hanoi, pp: 87-95. DOI: 10.1109/RIVF.2007.369140

Radhakrishna, A., A. Shaji, K. Smith, A. Lucchi and P. Fua *et al*., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Patt. Analysis Machine Intellig., 34: 2274-2282. DOI: 10.1109/TPAMI.2012.120

Stutz, D., 2014. Implementation of the superpixel algorithm called SEEDS.

Stutz, D., 2015. Efficient high-quality superpixels: SEEDS revised.

Turmer, S., 2014. Car detection in low frame-rate aerial imagery of dense urban areas. PhD Thesis, Institut für Photogrammetrie und Kartographie.

Van den Bergh, M., X. Boix and G. Roig, 2012. SEEDS: Superpixels extracted via energy-driven sampling. Proceedings of the European Conference on Computer Vision, Oct. 7-13, Florence, Italy, pp: 13-26. DOI: 10.1007/978-3-642-33786-4_2

Vedaldi, A. and S. Soatto, 2008. Quick shift and kernel methods for mode seeking. Lecture Notes Comput. Sci., 5305: 705-718. DOI: 10.1007/978-3-540-88693-8_52

Zhong, S. and J. Ghosh, 2003. A unified framework for model-based clustering. J. Machine Learn. Res., 4: 1001-1037.