Robust and Fast Detection of Moving Vehicles in Aerial Videos using Sliding Windows

Michael Teutsch and Wolfgang Krüger

Fraunhofer IOSB, Karlsruhe, Germany
{michael.teutsch, wolfgang.krueger}@iosb.fraunhofer.de

Abstract

The detection of vehicles driving on busy urban streets in videos acquired by airborne cameras is challenging due to the large distance between camera and vehicles, simultaneous vehicle and camera motion, shadows, or low contrast due to weak illumination. However, it is an important processing step for applications such as automatic traffic monitoring, detection of abnormal behaviour, border protection, or surveillance of restricted areas. In contrast to commonly applied object segmentation methods based on background subtraction or frame differencing, we detect moving vehicles using the combination of a track-before-detect (TBD) approach and machine learning: an AdaBoost classifier learns the appearance of vehicles in low resolution and is applied within a sliding window algorithm to detect vehicles inside a region of interest determined by the TBD approach. Our main contribution lies in the identification, optimization, and evaluation of the most important parameters to achieve both high detection rates and real-time processing.

1. Introduction

Cameras mounted on airplanes or Unmanned Aerial Vehicles (UAVs) are able to observe the ground area and collect video data in a highly effective and efficient way. Among the vast amount of potential applications are automatic traffic monitoring, detection of abnormal behaviour, border protection, or surveillance of restricted areas. These applications share the need for accurate detection and tracking of all moving objects inside the camera's field of view before the scene can be analyzed and interpreted. There are several aspects that complicate the automation of moving object detection such as the large distance between camera and objects leading to small-sized objects in the image, simultaneous object and camera motion, shadows, or low contrast due to weak illumination. Although many approaches for moving object detection in aerial video surveillance data exist in the literature, those methods are often lacking reliability, robustness, or real-time capability.

In this paper, we focus on the application of sliding windows [21] for vehicle detection in aerial videos. Originally developed for face and human detection [28, 5], this is a *brute force* or *exhaustive search* method used to localize objects of a certain class across the entire image. A classifier learns an object appearance model to reports its confidence about object existence at each search step. Several authors [2, 20, 26, 27] demonstrated the applicability of sliding windows for vehicle detection in aerial videos. We, however, aim to identify parameters that contribute most to both the detection performance and the runtime and optimize them to achieve high detection rates (reliability), few false positive (FP) detections (robustness), and real-time processing. Multiple object tracking can use these detections as input, but this is beyond the scope of this paper.

A track-before-detect (TBD) algorithm [24] is used in order to detect motion that is independent of the camera motion. As shown in Fig. 1, this independent motion is given by clustered motion vectors and does not represent vehicles, yet. As an alternative to TBD, difference images as applied in wide area surveillance with low frame rates of about 1 Hz can be used [22, 23, 30], but we process videos with high frame rates of 15–30 Hz, where difference images produce more noise compared to TBD [25]. Furthermore, difference images do not provide information about motion direction and velocity that we particularly use to reduce the search space of the sliding window. Not only can a large amount of FP detections be avoided this way, but also the processing time is reduced. Then, we discuss, evaluate, and optimize the most important sliding window parameters such as the choice of the vehicle appearance model, handling of variable object size, or optimization strategies. In urban scenes with up to 20 vehicles in the camera's field of view, we achieve detection rates of 88 % with only 2 % FP detections and processing times less than 40 ms per frame.

This paper is organized as follows: related work is discussed in Section 2. The processing chain for vehicle detection is described in Section 3. Parameters are evaluated and results are presented in Section 4. We conclude in Section 5.

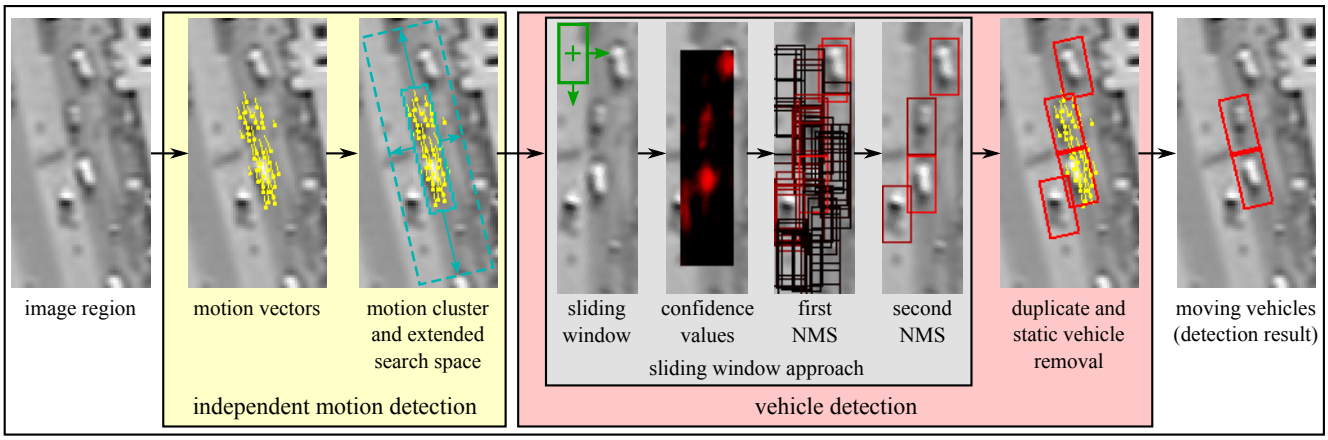

Figure 1. The processing chain for moving vehicle detection. A detailed description is given in Section 3.

2. Related Work

The sliding window approach is a popular method to combine object detection (localization) and recognition (classification) [21, 29]. A search window is shifted pixel by pixel in both horizontal and vertical direction across the entire image. At each window position, appearance features are extracted and a classifier returns a confidence value representing its certainty that the image region inside the window contains an object. The window size is fixed and, thus, only one classifier model is trained and used at M different image scales [5, 8] to detect objects of different sizes, where M is usually around 50 for human detection [1]. After the calculation of all confidence values, objects are detected by applying a Non-Maximum Suppression (NMS) to the confidence values and using a minimum classifier certainty threshold. In contrast to part based models [9, 19] which search for object parts and combine them to whole objects, the sliding window is a holistic object representation which models the object in its entirety. Holistic representation is usually better for small objects in the image as it is difficult to detect even smaller object parts reliably.

2.1. Vehicle Detection in Aerial Videos

Similar to Dalal's method [5], Türmer *et al*. [27] apply the sliding window approach with Histogram of Oriented Gradients (HOG) features and a Support Vector Machine (SVM) classifier to find stationary and moving vehicles. FP detections are rejected by using depth maps. Gaszczak *et al*. [12] detect vehicles using sliding windows with Haar features und cascaded AdaBoost which is very similar to the approach proposed by Viola and Jones [28]. Since vehicle orientation may vary, four discretized orientations are specified and one classifier is trained for each orientation. Nguyen *et al*. [20] use sliding windows with Haar features, orientation histograms, and Local Binary Patterns (LBP) as vehicle decriptors and AdaBoost for classification. Since multiple detections appear for each object due to the sliding window shift, mean-shift clustering is applied for a NMS. Cao *et al*. [2] propose a boosting light and pyramid sampling histogram of oriented gradients (bLPS-HOG) feature extraction method together with a linear SVM. Teutsch *et al*. [26] use Integral Channel Features (ChnFtrs) [8] and an AdaBoost classifier to detect vehicles in predetermined areas of independent motion.

2.2. Runtime Optimization

In order to achieve a high speed-up of sliding windows with stable object detection performance, several runtime optimization strategies have been proposed so far. A rather new approach is fast pre-scanning of the image in order to detect prominent edges and thus reduce the search space for the subsequent sliding window algorithm [4, 34]. Another idea is to speed up image rescaling by approximation: either nearby image scales are approximated while using one classifier model [7] or classifier decisions can be approximated across scales using few classifier models and only one image scale [1]. In order to avoid an exhaustive search, Gualdi *et al*. [13] propose to initialize sliding windows at random positions and follow the gradient of the classifier confidence values. Finally, a promising optimization for the AdaBoost classifier is achieved by using *soft cascades* [1, 7]. AdaBoost is a meta-algorithm for classification consisting of a cascade of weak classifiers. In a soft cascade, only few of the weak classifiers contributing most to the final decision are evaluated and the process stops as soon as a clear decision tendency emerges.

3. Moving Vehicle Detection

The processing chain for moving vehicle detection is visualized in Fig. 1. A small image region of 108×210 pixels is considered for this example. Separate modules are applied for independent motion detection and vehicle detection. These modules are described in more detail in the remainder of this section.

3.1. Independent Motion Detection

In order to detect independent motion, it is crucial to compensate the videos for camera motion first. We detect Harris corners [14] with sub-pixel accuracy and track them over time by a gradient based search in a local image region [24]. Corresponding corners between subsequent images are used to estimate homographies as global image transformations for image registration [15]. RANSAC is applied to remove outliers. We do not perform image alignment. Instead, velocities of tracks relative to the static background that exceed a threshold are assumed to originate from moving objects and are referred to as *(independent) motion vectors*. This is a TBD approach since we track objects in an environment where the entire scene seems to move and object motion is only a minor part [6, 10]. The motion vectors are clustered based on position, direction, and velocity. In Fig. 1, motion vectors are depicted in yellow color and motion clusters are represented by cyan rectangles. Vehicles that overtake each other or drive one behind the other often cause merged clusters while split clusters can be produced by large vehicles (trucks, busses) with weakly textured areas [26]. This raises the need for methods that are able to detect individual vehicles. As a first step, the motion vector clusters are extended in horizontal and vertical direction as shown in Fig. 1 in order to completely include even split clusters. These extended motion clusters define the search space (dashed cyan rectangle) for the sliding window.

3.2. Vehicle Detection

Extended motion clusters are rotated upright based on the direction of the related motion vectors. The assumption is that the orientation of a vehicle corresponds to its motion direction. This way, we achieve rotation invariance and need to apply the sliding window for only one orientation which is an important search space reduction. The scale of the entire scene can be normalized using the Ground Sampling Distance (GSD) that gives us the image resolution in meters. So, image rescaling is necessary only for different vehicle sizes and not for the distance between camera and scene.

3.2.1 Sliding Window Approach

In Fig. 1, the green rectangle represents the sliding window that is shifted pixel by pixel in both horizontal and vertical direction across the extended motion cluster. As the width of vehicles is usually smaller than their length, the window size is set to 16×32 pixels. At each window position, a classifier returns a confidence value representing its certainty that the current window contains a vehicle or not. This confidence value is stored at the center point of each sliding window position and depicted in Fig. 1. The light red color

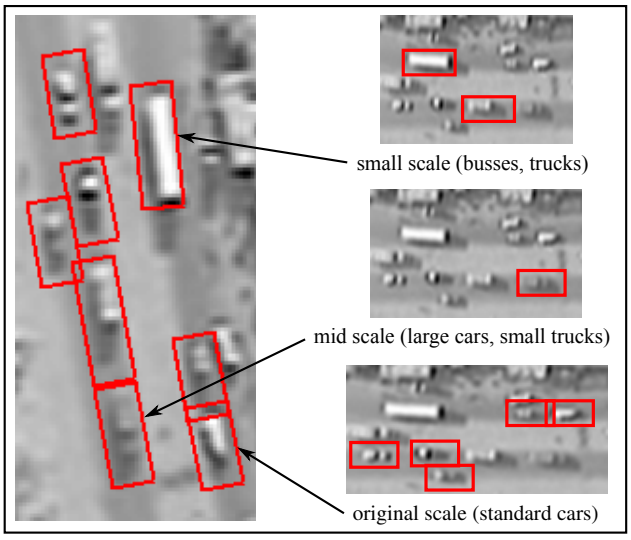

Figure 2. Instead of image rescaling with about 50 different scales for human detection using sliding windows [1], we use 3 scales inspired by typical vehicle sizes and GSD based normalization.

indicates a high certainty. Based on these confidence values, the first NMS rejects all positions of windows with lower certainty compared to their neighbors. Each remaining local maximum stands for one object hypothesis represented by a rectangle with size and position of the search window. If there is sufficient overlap with another local maximum, it is likely that these hypotheses originate from the same object. Then, a second NMS is applied to keep the hypothesis with the highest certainty and reject the others. Finally, a decision threshold T_d is used to reject weak local maxima. This process flow is inspired by the work of Dollár *et al.* [8]. The optimal value for T_d is derived from the precision-recall curves in Section 4.2.1 and thus determined experimentally.

Usually, about 50 different image scales are used for human detection in ground level images [1]. This is due to the highly variable size of persons in the image depending on the distance to the camera. As the scale of the scene in our aerial video data is normalized using the GSD, the number of different image scales can be significantly reduced. However, it has to be considered that the vehicle size can vary strongly: while the width of different vehicles is nearly constant, the length ranges between 4–5 m for a standard car and 15–20 m for busses or trucks. So, three different scales are introduced to the sliding window approach by keeping image width stable and varying image length as shown in Fig. 2. This is different compared to human detection where the ratio of width and length is fixed during image rescaling. By using only three different image scales, the search space for the sliding window is reduced. This is necessary since a vehicle model in top view and at low resolution of 16×32 pixels does not have the discriminative power of a person model in side view and at high resolution of 64×128 pixels. Since gradients are the most important information

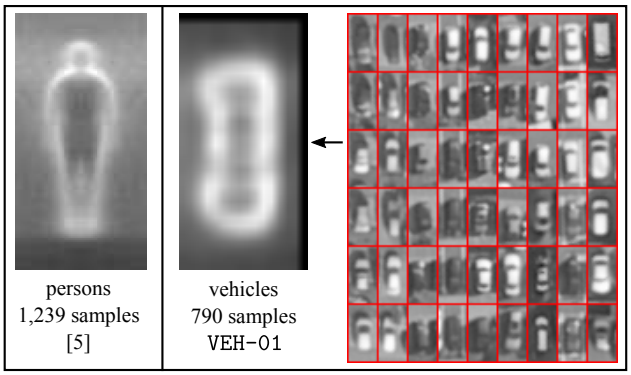

Figure 3. Mean gradient magnitudes: persons in side view at high resolution provide shape features for a classifier model of higher discriminative power than vehicles in top view at low resolution.

for modelling an object's shape, the average gradient magnitude images for persons [5] and our dataset VEH-01 (cf. Section 4.1) of vehicle samples for classifier training are depicted in Fig. 3. While head, shoulders, torso, and legs provide good features for a person model of high discriminative power, vehicles can be described only by their rectangular shape. The influence of such a weakly discriminative classifier model, image rescaling, and NMS thresholds to the robustness of the sliding window approach is discussed in Section 4.2.

3.2.2 Duplicate and Static Vehicle Removal

Extended motion clusters can overlap each other and thus cause the occurrence of duplicate detections after the application of the sliding window approach. Usually, duplicate detections significantly overlap each other and can be handled by keeping only the detection with the highest confidence value. Static vehicles inside the extended motion clusters are detected by the sliding window approach, too. As visualized in Fig. 1, they can be rejected by introducing a minimum threshold for the number of motion vectors inside each detection rectangle.

3.3. Runtime Optimization

Boosting is an attractive classifier for object detection with sliding windows for a number of reasons: (1) it offers good generalization performance [11], (2) it is able to perform feature selection during training [1], and (3) it is possible to achieve high frame rates during detection [1, 7] by using soft cascades [31].

During AdaBoost training a number of N weak classifiers c_n and weights α_n with $n \in \{1, \ldots, N\}$ are selected to build a strong classifier. The confidence value of the strong classifier for a sample x_k is computed as the weighted sum

$$s_k(N) = \sum_{n=1}^{N} \alpha_n c_n(x_k). \quad (1)$$

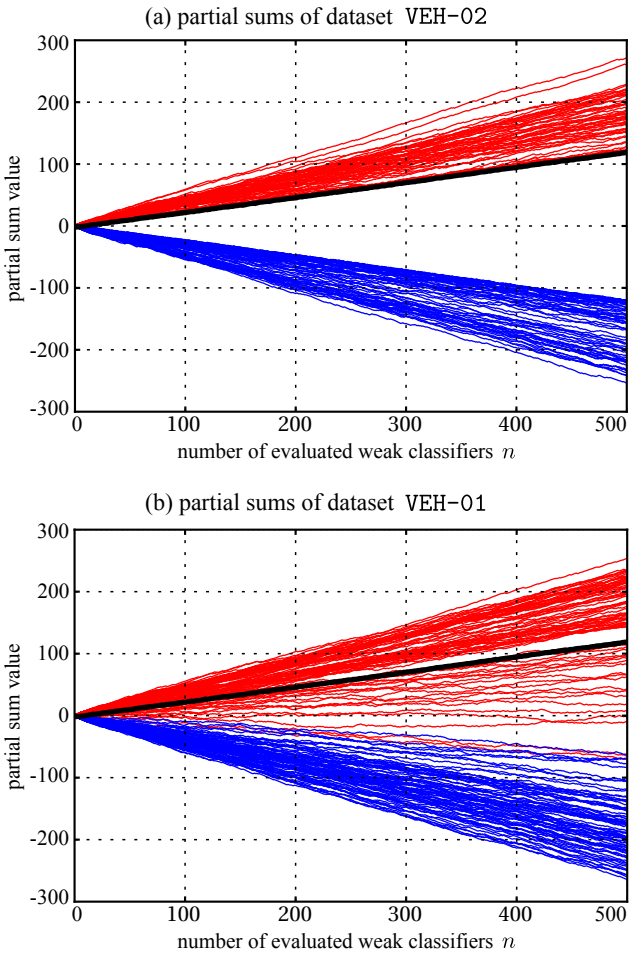

Figure 4. Partial sum values for positive (red) and negative (blue) samples of AdaBoost training dataset VEH-02 (a) and independent dataset VEH-01 (b). Choosing $T(n)$ (black) based on the training dataset [31] rejects too many positive samples in VEH-01.

The idea of runtime optimization by using soft cascades is to prune the search space for negative samples, i.e. detection windows containing no vehicles. Therefore, the computation of the classifier confidence value in Eq. 1 is terminated as soon as the value of the partial sum falls below a rejection threshold $T(n)$.

In order to find rejection thresholds $T(n)$, we use the direct backward pruning algorithm (DBP) proposed by Zhang and Viola [31]. First, a selection threshold $T(N)$ for the full sum is used to select positive samples (x_k, y_k) from the training set with $y_k = 1$ for positive samples and $y_k = -1$ for negative samples. Then, the rejection threshold $T(n)$ for each weak classifier index n is set to the minimum value of the corresponding partial sums for the previously selected positive samples:

$$T(n) = \min_{\{k | s_k(N) > T(N),\, y_k = 1\}} s_k(n), \quad (2)$$

$$\text{with } n \in \{1, \ldots, N\}. \quad (3)$$

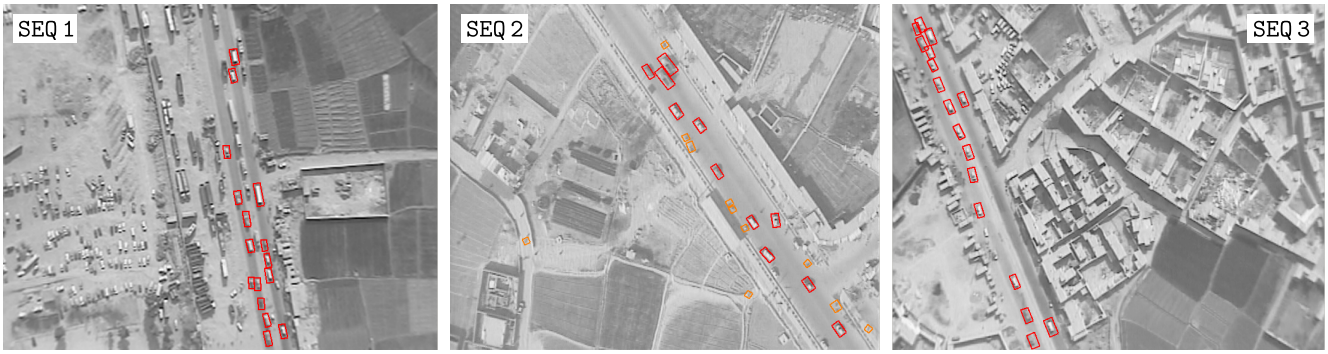

Figure 5. Example images taken from the three datasets SEQ 1, SEQ 2, and SEQ 3 used for our experiments. The ground truth (GT) for vehicles is visualized by red rectangles while other moving objects such as motorcycles or pedestrians are depicted in orange color.

Figure 4 (a) shows the partial sum values after AdaBoost training for dataset VEH-02 with $N = 500$ weak classifiers. Positive training samples (vehicle in detection window, red curve) were well separated from negative samples (no vehicle, blue curve). Applying DBP yields the minimum of the red curves as rejection thresholds $T(n)$ for each weak classifier index n. However, for our application we found that this approach leads to inferior results (cf. Section 4.2.3). The reason can be seen in Fig. 4 (b) which shows the partial sum values for the trained classifier on a different dataset VEH-01. The rejection thresholds determined with the original training data are too optimistic and will reject too many positive samples when applied to other datasets (overfitting).

As a consequence, we propose to use two training datasets: the first dataset is used to train the weak classifiers c_n and their weights α_n. The second dataset is then used to compute the partial sum curves for the trained classifier and to find the rejection thresholds with the DBP-algorithm.

In addition to pruning by soft cascades we also considered subsampling of sliding window positions in order to speed up the detection process. Finally, the number of weak classifiers used during training was reduced. The trade-off between detection quality and those runtime optimizations is investigated in Section 4.2.

4. Experiments and Results

In this section, we present the datasets used for our experiments, the parameter optimization, and the final results compared to other methods taken from the literature. Standard evaluation measures are considered such as false positives (FP), false negatives (FN), precision, recall, f-score, precision-recall curves [17] and Normalized Multiple Object Detection Precision (N-MODP) [18].

4.1. Datasets

Vehicle models, i.e. classifiers have to be trained before the sliding window can be applied. Therefore, negative samples of non-vehicles and positive samples of vehicles are necessary. The vehicles in two different wide area aerial images are manually labeled in order to generate the two training datasets. Each vehicle sample is cut out, rotated in upright position, and scaled to 16×32 pixels. Negative samples are generated in the same way at random positions in the background where no vehicles are visible. Each training dataset consists of a balanced number of positive and negative samples (approx. 1,300 samples per dataset). The two resulting datasets are denoted by VEH-01, VEH-02 and some samples of VEH-01 are depicted in Fig. 3.

The sliding window approach is evaluated with the three sequences SEQ 1, SEQ 2, and SEQ 3 that are shown in Fig. 5. They are coming from our own non-public data in top (nadir) view with a frame size of 720×576 pixels, a frame rate of 25 Hz, and a GSD around 0.3 m/pixel. A standard car covers about 7×15 pixels. The three sequences consist of 400, 200, and 450 gray-value frames with 4,731, 1,373, and 3,490 annotated moving vehicle samples. The ground truth (GT) was generated manually by tagging all moving objects with bounding rectangles rotated in motion direction. We only use GT for vehicles such as cars and trucks depicted in red color in Fig. 5 and skip other moving objects such as motorcycles or pedestrians (orange color). The reason is that we focus on moving vehicle detection and thus train the classifier for vehicle appearances only. It is possible to train separate models for persons or motorcycles in addition to the vehicle model, but since there is only little information available about the appearance of persons and motorcycles, the sliding window approach is prone to produce FP detections. The extraction of motion patterns is more promising in order to detect moving pedestrians at very low resolution [33].

4.2. Parameter Optimization

Sequence SEQ 1 is used in order to optimize the choice of the vehicle model, image rescaling, and runtime.

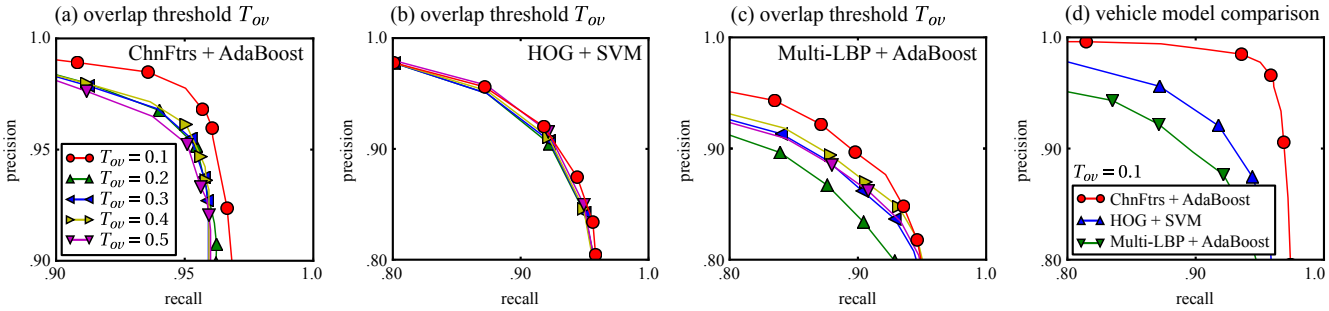

Figure 6. Optimization of the overlap threshold T_{ov} of the second NMS (a-c) and the choice of the vehicle model (d). The classifier decision threshold T_d is varied in order to present the results as precision-recall curves. ChnFtrs + AdaBoost with $T_{ov} = 0.1$ performs best.

4.2.1 Vehicle Model

A vehicle model in the context of this paper is the combination of a vehicle appearance descriptor and a classifier that exploits the descriptor and returns a confidence value. In order to find a well-fitting model for our data, we evaluate three models taken from the literature:

- ChnFtrs + AdaBoost [8]: Gradient magnitudes are subdivided in several orientation channels (we use 7). Each feature is a local sum of magnitudes in an area of random position and size in a random channel. These local sums are called *first-order features* [8] and give us better results compared to conventional Haar features. 2,000 of these features generate a feature pool for the AdaBoost feature selection.

- HOG + SVM [5]: HOGs extract edge information to decribe the object's shape. Best results are achieved with the parameters 8×8 blocks, 4 pixels block stride, and 9 histogram bins. The descriptor dimension is 756.

- Multi-LBP + AdaBoost [16]: LBPs are used to describe the object's texture. In four different quantizations, 8,192 LBPs are calculated to generate a large feature pool. We choose the parameters 8×8 pixels block size and 8 pixel block stride.

We consider these models as promising since both ChnFtrs + AdaBoost [20, 12, 26] and HOG + SVM [2, 27] have been successfully applied by other authors. Furthermore, since we do not want to focus on shape features only, texture features are analyzed by using Multi-LBP + AdaBoost. This approach performed well on low resolution gray-value images for human classification [16] and thus may be applicable to our data as well.

Classifier training is fully supervised using the datasets `VEH-01` and `VEH-02`. The parameters for optimization are (1) the choice of the vehicle model and (2) the overlap threshold T_{ov} of the second NMS (cf. Fig. 1) for each vehicle model. If the overlap of the rectangles of two detection hypotheses exceeds T_{ov}, then both detections are assumed to come from the same object and the weaker hypothesis is rejected. The choice of T_{ov} is crucial to prevent FP detections at vehicle shadows.

Table 1. Impact of image rescaling to the sliding window. The different rescaling strategies are described in Section 4.2.2.

vehicle model	rescaling strategy	evaluation measures		
		FP	FN	f-score
HOG + SVM	strategy 1	889	245	0.888
	strategy 2	780	258	0.896
	strategy 3	190	606	0.912
ChnFtrs + AdaBoost	strategy 1	357	230	0.939
	strategy 2	283	244	0.945
	strategy 3	83	268	0.962

Each classifier returns different confidence values and a comparison between classifiers is difficult as range and scale of these values usually do not fit. Variation of the decision threshold T_d and plotting the resulting values for precision and recall to common graphs as seen in Fig. 6 is a way to overcome this problem. The graphs (a-c) show the optimization of T_{ov} for the three vehicle models individually. Each model achieves the best performance for $T_{ov} = 0.1$. In Fig. 6 (d), the three vehicle models are compared to each other. ChnFtrs + AdaBoost clearly outperforms HOG + SVM and MultiLBP + AdaBoost. One explanation for this performance difference is given by Benenson *et al.* [1]: the automatic feature selection of AdaBoost applied to the randomly positioned ChnFtrs is superior compared to hand designed descriptors such as HOG and Multi-LBP. While all HOG and Multi-LBP blocks are located at fixed positions with fixed size, the AdaBoost classifier chooses features with largest discriminative power without such limitations. These features can be located between blocks and do not have fixed size. So, ChnFtrs may be more tolerant to small shift, deformation, and size variation of vehicles inside the sliding window.

4.2.2 Image Rescaling

In this experiment, the influence of image rescaling to the performance of the sliding window approach is analyzed. As already mentioned in Section 3.2.1, the classifier model

Table 2. Optimization of soft cascade parameters $T(n)$ and $T(N)$ as well as sliding window subsampling with 2 by 4 pixel shifts.

dataset used to find $T(n)$	$T(N)$	FP	FN	f-score	$\bar{n}_e$
-	-	91	262	0.962	500.0
VEH-02	0	15	992	0.881	18.5
VEH-01	0	91	263	0.962	70.3
VEH-01	20	90	265	0.960	61.5
VEH-01	50	80	292	0.950	46.2
subsampling with 2 by 4 pixel shifts					
VEH-01	0	94	251	0.963	73.4
VEH-01	20	80	303	0.959	64.2
VEH-01	50	46	441	0.946	48.1

for vehicles in top view does not have high discriminative power and, thus, the number of FP can increase faster than the decrease of FN when using many image scales. This is demonstrated in Table 1. For both HOG + SVM and ChnFtrs + AdaBoost, three different cases of image rescaling are evaluated. Strategy 1 is the baseline approach as it is inspired by image rescaling for human detection: eleven different scale factors s_i are used to rescale the extended, upright motion cluster of width w_0 and height h_0. For the i-th rescaled motion cluster, width $w_i = s_i \cdot w_0$ and height $h_i = s_i \cdot h_0$ are rescaled jointly. In strategy 2, width $w_i = w_0$ is constant while only height $h_i = s_i \cdot h_0$ is rescaled by eleven scale factors s_i. Finally, strategy 3 is the proposed approach visualized in Fig. 2 and similar to strategy 2 but with three scale factors instead of eleven. Rescaling only the height of the motion cluster outperforms the baseline approach, but even larger improvement is achieved by using only three different scales. This is visible for both HOG + SVM and ChnFtrs + AdaBoost. The application of more scale levels reduces the number of FN but increases the number of FP much more at the same time since more shadows or other rectangular non-vehicle appearances are detected.

4.2.3 Runtime Optimization

Our first step in order to speed up the detection process is to train an AdaBoost soft cascade (cf. 3.3). We aim to reduce the average number of evaluated weak classifiers $\bar{n}_e$. Table 2 shows the detection performance of different soft cascades on test sequence SEQ 1. In all variants, dataset VEH-02 is used to train the weak classifiers and their weights. The first row in Table 2 shows the results for the full cascade. The DBP-algorithm is used to find the rejection thresholds $T(n)$ for pruning. Using the original training set VEH-02 for DBP gives inferior results (row 2). This was already demonstrated in Fig. 4 for the vehicle datasets VEH-01 and VEH-02 and is confirmed here for the sliding window approach after motion detection applied to SEQ 1. Using a second training set VEH-01 for DBP leads to much better detection results (row 3 to 5) at the cost of less pruning (larger $\bar{n}_e$ compared to row 2).

Table 3. Optimization by reducing the number of weak classifiers N for the soft cascade highlighted in red color in Table 2.

N	FP	FN	f-score	$\bar{n}_e$
500	94	251	0.963	73.4
200	98	289	0.958	37.5
100	79	376	0.950	22.3

Table 4. Overall runtime evaluation.

method	runtime
independent motion detection and clustering ($\sim$ 22,000 motion vectors per image)	24.2 ms
sliding window approach ($\sim$ 10–15 motion clusters per image)	11.0 ms
duplicate and static vehicles removal	1.5 ms
entire processing chain	36.7 ms

The next step is to consider subsampling of sliding window positions. Instead of shifting the sliding window pixel by pixel across the image, we found that shifting in steps of 2 pixels horizontally and 4 pixels vertically is still able to achieve good detection performance as seen in the second part of Table 2. The row highlighted in red color represents the set of parameters that we propose: $T(N) = 0$ and subsampling with 2 by 4 pixel shifts.

With this choice of parameters, we finally aim to reduce the number of weak classifiers N used during initial AdaBoost training. Table 3 shows the results for three different soft cascades trained on datasets VEH-02 (AdaBoost) and VEH-01 (DBP). A good trade-off between detection performance and runtime optimization is given with a number of 200 weak classifiers.

In Table 4, we present the runtime for the selected soft cascade (marked red in Table 3). Independent motion detection takes about 24.2 ms per frame on average and produces about 22,000 motion vectors. This process is already optimized, but the optimization strategy is not discussed in this paper. The sliding window approach runs 11 ms per frame where the ChnFtrs descriptor is calculated in about 5.3 ms and the classification process needs 5.7 ms. Without optimization, classification takes around 712 ms per frame. So, we achieved a speedup by factor 125 while the detection rate deteriorates only slighty. Overall, the runtime is 36.7 ms per frame for the most crowded sequence SEQ 1 using a standard PC with a 3.4 GHz Intel Quad Core i7 CPU. However, so far we did not consider parallel processing of the motion clusters which can achieve further reduction of the runtime.

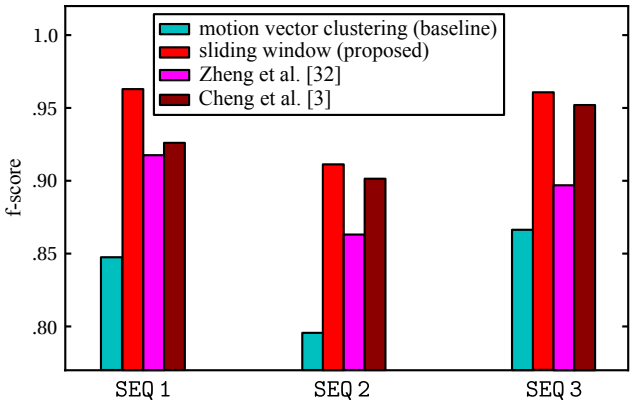

Figure 7. The proposed sliding window approach outperforms methods for blob [32] and contour based [3] object segmentation.

4.3. Final Results

In order to demonstrate the effectiveness of the sliding window approach compared to methods taken from the literature, we evaluate detection approaches based on object segmentation. We use the same TBD algorithm with extended motion clusters as search space and apply one algorithm for blob extraction based on the tophat transform [32] and one algorithm for edge based contour extraction based on clustering of Canny edges and Harris corners [3]. Each blob or cluster is considered as one detected vehicle. The authors of the second method also propose to perform color segmentation and fuse the information in a Bayesian network. As we do not have color information in our sequences SEQ 1, SEQ 2, and SEQ 3, we skip these processing steps in our evaluation. The detection performance is compared using the f-score as visualized in Fig. 7. Motion vector clustering is considered as baseline approach and is clearly improved by all three methods. Inner vehicle structures such as trunks or engine hoods cause split detections (i.e. FP detections) for blob and contour based segmentation. Merged detections (i.e. FN detections) often occur in SEQ 1 and cause the large gap between the sliding window approach and the segmentation methods. We also evaluate the average overlap between detection and ground truth rectangles. This is given by the N-MODP that lies between 0.6 and 0.7 for the sliding window and between 0.5 and 0.6 for the segmentation methods which suffer from undersegmentation due to street texture or sidewalk edges.

5. Conclusions

The sliding window approach is a well suited method for vehicle detection in aerial videos. In our experiments, we show that it is able to outperform detection algorithms based on object segmentation especially in urban scenes with many vehicles driving on busy streets. Parameters of the sliding window approach that contribute most to the detection and processing performance are identified and optimized: we propose (1) to use ChnFtrs + AdaBoost as vehicle model, (2) to rescale the image with only three different scales and only in width direction with fixed height, and (3) to optimize the runtime with soft cascades, subsampling, and reducing the number of weak classifers in the AdaBoost model. In this way, we achieve detection rates of 88 % with only 2 % of FP detections across different datasets and an average processing time less than 40 ms per frame on standard hardware in scenes with up to 20 moving vehicles. The low FP rate together with detection confidences provided by the classifier make sliding window based object detection suitable for a combination with multiple object tracking approaches that rely on initial detections.

Acknowledgments

The research reported in this contribution was funded by the Technical Center for Information Technology and Electronics (WTD 81) of the German Federal Office for Armament and Procurement (BAAINBw).

References

[1] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 3, 4, 6

[2] 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*, 21(10):1522–1533, Oct. 2011. 1, 2, 6

[3] H.-Y. Cheng, C.-C. Weng, and Y.-Y. Chen. Vehicle Detection in Aerial Surveillance Using Dynamic Bayesian Networks. *IEEE Transactions on Image Processing*, 21(4):2152–2159, Apr. 2012. 8

[4] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[5] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. 1, 2, 4, 6

[6] S. J. Davey, M. G. Rutten, and B. Cheung. A Comparison of Detection Performance for Several Track-Before-Detect Algorithms. In *Proceedings of the 11th International Conference on Information Fusion (FUSION)*, 2008. 3

[7] P. Dollár, S. Belongie, and P. Perona. The Fastest Pedestrian Detector in the West. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2010. 2, 4

[8] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009. 2, 3, 6

[9] 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*, 32(9):1627–1645, Sept. 2010. 2

[10] U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception. In W. G. Kropatsch, R. Sablatnig, and A. Hanbury, editors, *Pattern Recognition*, volume 3663 of *Lecture Notes in Computer Science (LNCS)*, pages 216–223. Springer Berlin / Heidelberg, Sept. 2005. 3

[11] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, Aug. 1997. 4

[12] A. Gaszczak, T. P. Breckon, and J. Han. Real-time people and vehicle detection from UAV imagery. In *Proceedings of SPIE Vol. 7878, Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques*, 2011. 2, 6

[13] G. Gualdi, A. Prati, and R. Cucchiara. Multistage Particle Windows for Fast and Accurate Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1589–1604, Aug. 2012. 2

[14] C. Harris and M. Stephens. A combined corner and edge detector. In *In Proceedings of the Fourth Alvey Vision Conference*, 1988. 3

[15] R. Hartley and A. Zisserman. *Multiple-View Geometry in Computer Vision*. Cambridge University Press, 2 edition, Mar. 2004. 3

[16] C. Heng, S. Yokomitsu, Y. Matsumoto, and H. Tamura. Shrink boost for selecting multi-LBP histogram features in object detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 6

[17] N. Japkowicz and M. Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011. 5

[18] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, M. Boonstra, V. Korzhova, and J. Zhang. Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):319–336, 2009. 5

[19] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77:259–289, May 2008. 2

[20] T. T. Nguyen, H. Grabner, H. Bischof, and B. Gruber. Online Boosting for Car Detection from Aerial Images. In *Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future (RIVF)*, 2007. 1, 2, 6

[21] C. Papageorgiou and T. Poggio. A Trainable System for Object Detection. *International Journal of Computer Vision (IJCV)*, 38(1):15–33, June 2000. 1, 2

[22] V. Reilly, H. Idrees, and M. Shah. Detection and Tracking of Large Number of Targets in Wide Area Surveillance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. 1

[23] I. Saleemi and M. Shah. Multiframe Many-Many Point Correspondence for Vehicle Tracking in High Density Wide Area Aerial Videos. *International Journal of Computer Vision (IJCV)*, 104(2):198–219, Sept. 2013. 1

[24] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994. 1, 3

[25] M. Siam and M. ElHelw. Robust Autonomous Visual Detection and Tracking of Moving Targets in UAV Imagery. In *Proceedings of the IEEE International Conference on Signal Processing (ICSP)*, 2012. 1

[26] M. Teutsch, W. Krüger, and J. Beyerer. Evaluation of Object Segmentation to Improve Moving Vehicle Detection in Aerial Videos. In *Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2014. 1, 2, 3, 6

[27] S. Türmer, 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 (JSTARS)*, 6(6):2327–2337, Dec. 2013. 1, 2, 6

[28] P. Viola and M. Jones. Robust Real-time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 1, 2

[29] Y. Wei and L. Tao. Efficient Histogram-Based Sliding Window. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2

[30] J. Xiao, H. Cheng, H. Sawhney, and F. Han. Vehicle Detection and Tracking in Wide Field-of-View Aerial Video. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 1

[31] C. Zhang and P. A. Viola. Multiple-Instance Pruning For Learning Efficient Cascade Detectors. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1681–1688. Curran Associates, Inc., 2008. 4

[32] Z. Zheng, G. Zhou, Y. Wang, Y. Liu, X. Li, X. Wang, and L. Jiang. A Novel Vehicle Detection Method With High Resolution Highway Aerial Image. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS)*, 6(6):2338–2343, 2013. 8

[33] J. Zhu, O. Javed, J. Liu, Q. Yu, H. Cheng, and H. Sawhney. Pedestrian Detection in Low-resolution Imagery by Learning Multi-scale Intrinsic Motion Structures (MIMS). In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5

[34] C. L. Zitnick and P. Dollár. Edge Boxes: Locating Object Proposals from Edges. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 2