Experimental evaluation of vehicle detection based on background modelling in daytime and night-time video

Igor Lipovac, Tomislav Hrkać, Karla Brkić, Zoran Kalafatić and Siniša Šegvić
Faculty of Electrical Engineering and Computing
Email: name.surname@fer.hr

***Abstract*—Vision-based detection of vehicles at urban intersections is an interesting alternative to commonly applied hardware solutions such as inductive loops. The standard approach to that problem is based on a background model consisting of independent per-pixel Gaussian mixtures. However, there are several notable shortcomings of that approach, including large computational complexity, blending of stopped vehicles with background and sensitivity to changes in image acquisition parameters (gain, exposure). We address these problems by proposing the following three improvements: (i) dispersed and delayed background modeling, (ii) modeling patch gradient distributions instead of absolute values of individual pixels, and (iii) significant speed-up through use of integral images. We present a detailed performance comparison on a realistic dataset with handcrafted groundtruth information. The obtained results indicate that significant gains with respect to the standard approach can be obtained both in performance and computational speed. Experiments suggest that the proposed combined technique would enable robust real-time performance on a low-cost embedded computer.**

I. Introduction

In this paper we consider vehicle detection at urban intersections. Traditionally, this problem has been solved by inductive loop sensors capable of detecting the presence of a vehicle. However, applying inductive loops for vehicle detection is expensive, primarily due to the need to conduct construction works and to stop the traffic during the installation as well as for maintenance. Therefore, the traffic management companies search for alternative solutions which would enable easy sensor replacement and maintenance. Computer vision techniques are very suitable for this task.

The usual computer vision scenario involves a fixed-view camera above the road and suitable algorithms to detect moving objects of appropriate size. This leads us to the well known problem of background modeling for which numerous solutions have been proposed. For the application scenario which involves day and night video capture, it is necessary to have an adaptive background model. Another constraint that should be taken into account is that the target system should be suitable for mounting at road infrastructure elements. Therefore, it would be beneficial to develop algorithms that could run on embedded hardware, which would also significantly reduce the installation costs.

We first considered the baseline background modeling approach based on per-pixel Gaussian mixtures, and evaluated it in the typical urban intersection scenario. This preliminary evaluation identified several important problems.

1) during the red light phase, the vehicles stop for a relatively long period and due to the background adaptation tend to influence the background model;
2) the classical running average model with exponential learning curve [12] tends to overemphasize the influence of the waiting vehicles;
3) automatic camera adaptation causes significant changes of the image in some situations, leading to miss-detections;
4) the detection of the vehicles of the color similar to the color of the road is often unsuccessful.

In order to address these problems we evaluate several improvements to the baseline background modelling approach. Firstly, we delay the model used for object detection in order to reduce the influence of waiting cars to their own detection. The idea here is to use the background model built before the arrival of the stopped cars and thus avoid using an infected model. Secondly, we attempt to reduce the influence of waiting cars by introducing a more appropriate weighting of the incoming frames through a two-stage background model. Thirdly, we attempt to reduce the dependence on absolute pixel values by building gradient-based background models. In order to improve the resistance to noise and at the same time reduce computational complexity, we refrain from considering individual pixels and instead model the gradient distribution above an overlapping set of background patches.

II. Related work

Computer vision-based approaches to the estimation of traffic flow parameters have been the subject of a lot of recent research. A common approach to separate foreground objects from the background scenery is based on background modelling. In such approaches, a statistical model that describes the background state of each pixel is constructed and subsequently compared to the current video frame. Pixels in which the difference is significant are considered to belong to foreground objects.

A number of methods for background model construction has been proposed. Especially popular have been time-adaptive Gaussian mixture models [14], [12], [15]. In these methods, each pixel is represented with a set of weighted Gaussian

distributions. Based on the assumption that background is visible most of the time in each pixel position, distributions are ordered according to their weights, and those more relevant are considered to model the background, while the remaining model the foreground. The per-pixel models are updated with each new observation, with older observations losing influence over time.

A comparison of several different background subtraction algorithms for detecting moving vehicles and pedestrians in urban traffic video sequences is given in [4]. The tested algorithms are classified as non-recursive (including simple frame differencing, median filtering, linear predictive filtering and non-parametric estimate of the pixel density function) or recursive (approximated median filter, Kalman filter and mixture of Gaussians). The evaluation is performed on four different video sequences with manually annotated moving objects in ten frames in each sequence as a ground truth. The algorithms are then evaluated by measuring precision and recall for different algorithm parameters. Mixture of Gaussians produces the best results, but median filtering offers a simple alternative with competitive performance. Frame differencing produces significantly worse results than all the other schemes.

Herrero and Bescós [9] provide another detailed overview and an evaluation of commonly used background subtraction techniques. The approaches covered by the overview are divided into simple (frame differencing, running average, median filtering), unimodal (Gaussian or chi-square modelling) and multimodal (mixtures of Gaussians, mean-shift algorithm, kernel density estimation and hidden Markov models). Evaluation is performed on video sequences from the dataset introduced in [13], obtained by combining separately recorded foreground and background videos, so the segmentation masks are known. The evaluation results suggest that chi-square modelling performs best in most scenarios. However, the authors note that mixtures of Gaussians and simple median filtering performed especially well in cases of highly dynamic backgrounds. Overall, the experimental findings in the evaluation supports the notion that relatively good results can be obtained with very simple techniques.

A more recent evaluation of bacground subtraction techniques with an emphasis on video surveillance is given by Brutzer et al. [3]. Nine background subtraction methods are compared at the pixel level. To alleviate the problem of ground truth collection, the authors rendered complex artificial scenes that address several challenges in background subtraction: gradual and sudden illumination changes, dynamic background, objects similar to background, shadows, initialization with foreground objects present and noise. The top-performing method is ViBe, a method proposed by Barnich and Van Droogenbroeck [1]. ViBe introduces several interesting innovations, e.g. storing a history of actual pixel values for a given pixel instead of building a statistical model, having a random update policy, doing background initialization from a single frame by assuming that neighboring pixels share a similar temporal distribution, etc.

Most background techniques assume a single rate of adaptation that determines how adaptive the model is to the change in pixel value. However, this can be inadequate in scenes such as traffic intersections, where objects move at a variety of speeds. A fast-adapting algorithm can miss detection of parts of homogeneous moving objects, since they quickly become part of the background. On the other hand, slow adapting algorithm leave long trails (”ghosts”) behind initially stationary objects that suddenly start to move, such as cars waiting at the crossroad. Algorithms with slow adaptation rate are also more sensitive to sudden global illumination changes. To cope with this, Cheung and Kamath [5] propose a dual-stage algorithm that first builds a foreground mask using a slow-adapting Kalman filter, and then validates individual foreground pixels by a simple moving object model, built using foreground and background statistics as well as the frame difference.

Another approach was suggested by Harville [8]. He proposed a framework for guiding evolution of pixel-level mixture of Gaussians models by using feedback from higher-level modules, such as module for person detection and tracking, or module for detection of rapid changes in global illumination, camera gain or camera position. The feedback of each module can be classified either as positive, which serves to enhance correct foreground segmentation, or as negative, which aims to adjust the pixel-level background model in order to prevent the re-occurrence of detected foreground mistakes.

To improve the robustness of vehicle detection against illumination changes and small camera movements, as well as the ability to track vehicles in case of occlusions and crowded events, Batista et al. [2] propose a dual-stage approach consisting of pixel-level and block-level stages. The pixel-level stage uses a multi-layered and adaptive background modeling, based on three image models. Two of them are used to model the dynamics of the background allowing the system to cope with intensity variations, while the third is used in the cleaning/validation process, being a direct copy of the past image. The block-level stage performs a 8x8 block-region analysis to label the blocks belonging to different vehicles and track them over a stack of images.

As a part of the University of South California Clever Transportation Project, Kim et al. [10] propose a system for real-time traffic flow analysis. The system aims to replace traffic loop detectors with cameras utilizing computer vision techniques. A special coprocessor, the Viewmont video analytics coprocessor, has been provided by Intel, who is a partner on the project. The coprocessor is specifically tailored toward video processing, enabling significant speed-up when compared to a conventional CPU. At the time of writing there is no information about the coprocessor on Intel's webpage, and it does not seem to be commercially available. In order to use the system, one needs to define a region of interest where the traffic is most visible, and within it a series of virtual lines spanning across individual lanes. Background is subtracted using frame averaging, and moving objects are extracted. Morphological operations are applied to obtain crisp boundaries of the moving objects and remove noise. Passing

of the vehicles is detected by counting the relative proportion of pixels belonging to moving objects crossing a virtual line to the total number of pixels comprising the line. In the evaluation, the results obtained by the system are compared to the output from real loop detectors. The main two identified problems are dense traffic and vehicle shadows.

III. The standard approach and its shortcomings

All background modelling approaches assume that each particular image pixel in most video frames is projected from the background scenery. Thus, a fair estimate of the actual background should be obtainable by some kind of an average pixel value across many frames. By comparing the actual value with the estimated model a pixel can finally be classified into either the background or the foreground class.

Pixel averaging is usually achieved by fitting a Gaussian mixture model with few components (n=2 or n=3) to each individual pixel over a number of frames. Multiple components are useful since they can account for small camera motions due to vibrations. The recovered variances allow to perform the classification by taking into account the actual camera noise.

In order to avoid the need for storing a large number of video frames, the standard background modelling approach estimates the required per-pixel Gaussian mixtures by employing the exponential moving average. In this approach, the evolution of the single component model (μ, σ) at the pixel (x, y) of the current image I_k can be described with the following equations (note that the free parameter α regulates the adaptivity of the model).

$$\mu_k[x,y] = \alpha I_k[x,y] + (1-\alpha)\mu_{k-1}[x,y] \ , \tag{1}$$

$$\sigma_k^2[x,y] = \alpha(I_k[x,y] - \mu_k[x,y])^2 + (1-\alpha)\sigma_{k-1}^2[x,y] \ . \tag{2}$$

These equations are easily extended to the multi-component case by weighting the contribution of the current pixel with the distances from the component centers [15]. The model needs to be initialized on a suitable video sequence either by straight-forward Gaussian fitting (one component) or by the EM algorithm (multiple components).

IV. The proposed improvements to the baseline approach

Unfortunately, it is very difficult to find the parameter α of the standard approach (2) which achieves an optimal ballance between robustness to stopped objects and adaptivity to daily illumination changes. If α is too large, stopped cars start disturbing the model. If α is too small, the model will not be adaptive enough to follow illumination changes due to meteorological conditions. This could be improved by storing a history of values for each given pixel [1] and calculating the correct running average:

$$\mu_k[x,y] = \sum_{i=k-N}^{k-1} I_i[x,y] \ , \tag{3}$$

$$\sigma_k^2[x,y] = \sum_{i=k-N}^{k-1} (I_i[x,y] - \mu_k[x,y])^2 \ . \tag{4}$$

However, we refrain from that approach due to memory requirements which would be difficult to meet on a low-cost embedded computer. The standard approach is also very sensitive to global changes of the camera acquisition parameters, which occur whenever large white or black vehicle enter the field of view. Finally, the standard approach makes it difficult to process more than one video stream on a low-cost embedded computer. These shortcomings shall be addressed in the following subsections.

A. Two stage background model

In order to deal with long term illumination changes and stopped cars becoming part of the background model after a prolonged period of waiting for the traffic light change, we propose the following background modelling approach. Specific to this approach is that we build two background models and that is why we call it the two stage approach. The first model in our two stage approach is the baseline background model and it is updated with every frame of the video. The second model is refreshed every N frames with the representation from the first model that is $2N$ frames old. This way we disperse and delay the contribution of the images that were used for updating the first model and hopefully we create a model that is more robust and deals better with aforementioned problems. Also we keep our model adaptive to long term changes and we do not lose information because of the first stage model that is updated with every frame. Both single stage (baseline approach) and two stage model use exponential running average to update with the current frame. Each frame contribution in both single stage and two stage background model is discussed and presented.

The frame contribution in the standard model (2) features the contribution $C_k^{(1)}$ in the frame k:

$$C_k^{(1)} = \alpha \ , \tag{5}$$

$$C_{k-1}^{(1)} = \alpha(1-\alpha) \ , \tag{6}$$

$$C_{k-2}^{(1)} = \alpha(1-\alpha)^2 \ . \tag{7}$$

Let the index of the current frame again be given by k. Then the frame contribution of the two stage dispersed and delayed model in the frame i can be expressed in terms of the contribution of the standard one-stage model $C_i^{(1)}$ and the update parameter β:

$$C_i^{(2)} = \sum_{j=0}^{\lfloor \frac{k-i-1}{N} \rfloor} \beta(1-\beta)^j \cdot C_{i+j\cdot N}^{(1)} \tag{8}$$

The two contribution models are shown in Figure 1. In comparison with the standard model (left), in the two stage model the frame contribution is dispersed and the domination of the most recent frames in the final contribution is reduced.

B. Addressing sudden changes of pixel brigthness

Sudden changes of pixel brigthness may occur either due to non-linear illumination variations such as clouds (dis-)occluding the sun or vehicle lights, or due to automatic

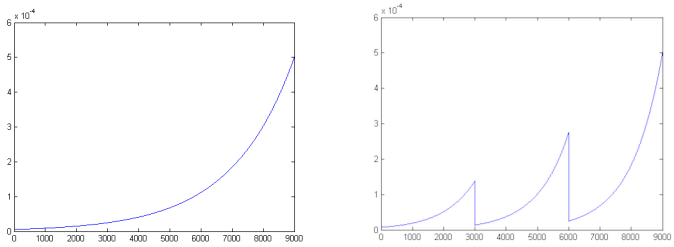

Fig. 1. Background model contribution depending on the frame number in the standard approach (left, $\alpha = 0.001$), and in the proposed two stage approach (right, $\alpha = 0.001$, $\beta = 0.5$, N=3000).

adaptation of acquisition parameters (gain, shutter). The latter usually occur due to large vehicles with extremely light or extremely dark colours. Background models based on absolute pixel values do not deal very well in those situations because absolute pixel values can change significantly and cause false positive detections to occur. We have considered two different approaches to deal with this problem by designing background models with built-in invariance to absolute pixel brightness.

1) Modeling the gradient distributions: The simplest way to achieve invariance to absolute pixel brightness is to design a model working on image gradient. Image gradient is tricky to work with, since it is not informative in flat regions. Hence we aggregate the gradient information over image patches and represent its distribution in the form of a histogram of oriented gradients (HoG) [6]. Finally, we build the background model consisting of a number of such histograms, recovered over a collection of image patches.

In the implementation, we first divide the image area under the virtual inductive loops into cells grouped in overlapping blocks. Then we calculate the HoG descriptors for each cell and create a normalized feature vector from histogram values for each block. Since the blocks are overlapping, each cell contributes to more than one block. This way we get a model that is more invariant to sudden changes of local brightness, colour contrasts and shadow appearance.

2) Structure-texture decomposition: This approach is based on the image denoising algorithm [11], [7] which represents the input image f as a combination of structure u (representation at the coarse scale) and texture α (additive noise and fine details), as shown in Equation (9).

$$f = u + \eta \tag{9}$$

Performing this denoising process on images with a low amount of additive noise suppresses the texture component of an image and leaves the structure of the image intact. We get the texture component η by subtracting the structure component u from the original image f and use it to build a brightness invariant background model.

C. Speeding up frame processing

One of our goals was to achieve real-time detection system that could run on an embedded computer with a low performance processor. In order to accomplish this we introduce HoG calculation using integral image which is a data structure and algorithm for quickly and efficiently calculating the sum of values in a rectangular region of interest inside the image. We calculate gradients for each cell in a single pass over the image and once the integral image has been computed we can calculate the value of a specific rectangular cell in a constant time and that results in a significant speed-up over the straightforward MoG approach. Frame processing time is measured and presented in the following section, Table III.

V. Experiments

In order to evaluate the proposed detection models we have collected three realistic videos acquired at real urban intersections. Hyperparameters of the considered detection models are first trained by exhaustive grid-search on validation subsets. The detection models with best hyperparameters are finally evaluated on independent test subsets.

A. Datasets

We have performed experiments using three test videos taken by a camera placed on top of a traffic light post (cf. Fig. 2). The video footage from the first video is is taken during daylight hours and it is far better than the other two in terms of image quality and camera stability. The first video consists of 27500 frames showing one urban intersection. The background models are initially trained on the first 10000 frames. Of the remaining 17500 frames, 7500 were used for validation, 10000 for testing and every fifth frame was manually marked as a validation or a test sample. Therefore, the validation subset consists of a total of 1500 images and the test set contained 2000 images. The second and the third videos are acquired from a different location by a low cost camera producing images with a lot of noise. Additionally, there is significant background motion caused by the wind shaking the camera. The second video is taken during afternoon, and it consists of 27500 frames divided in 10000 frames for pre-training of the background model, 7500 for validation and 1000 for testing. The third video is taken during night-time and it consists of 37500 frames divided in 10000 frames for pre-training, 10000 for validation and 17500 for testing.

Fig. 2. Representative frames from the dataset 1 (left), dataset 2 (middle) and the dataset 3 (right).

The following rules were followed when annotating images in all three videos.

1) Frames in which the vehicle is completely covering the virtual loop area were marked as positives.
2) Frames in which the loop is completely empty were marked as negatives.
3) Frames where the loop area is partially covered were discarded.

Using this annotation policy, we have effectively reduced our first dataset to 1086 validation and 1561 test images, the second dataset was reduced to 1302 validation and 1781 test images and the third to 1365 validation and 3128 test images.

B. Validation

The next phase in experiments included validating parameters for each of the proposed background models. We decided to validate single stage MoG and HoG models, delayed MoG model and finally two stage dispersed and delayed MoG and HoG models. For each parameter affecting these background models we defined a definite set of possible values and performed validation tests using grid search optimisation approach to evaluate every possible detection model with a certain combination of parameter values. When validating standard models, the parameters taken into consideration were the learning rate, pixel difference threshold in MoG models for determining foreground pixels, maximum number of Gaussian mixtures in a MoG model and the difference threshold for a single cell needed to indicate the detection in HoG models. When validating dispersed and delayed models we introduce the learning rate of the short term standard model, the dispersed and delayed model learning rate and the number of frames used for building our short term model. We were interested in optimisation of the aforementioned parameters and wanted to determine at which level each of these parameters affects our results.

We evaluated detection model performance by measuring its precision and recall and calculating the average precision AP as the area under the precision-recall curve. If we denote the precision and recall in k-th data point by $P(k)$ and $r(k)$, then the AP can be determined as follows.

$$AP = \sum_{k}^{n} P(k)\Delta r(k) \tag{10}$$

Our reference parameter for drawing precision-recall curves was percentage of virtual loop covered by the vehicle for MoG models, and cell detection threshold for HoG models. For each considered approach we have found the best set of hyperparameters by exhaustive search on all three validation subsets. The identified best detection models are evaluated on independent test sets, as presented in the next section.

C. Testing

The six considered detection approaches are evaluated on the three independent test subsets. The obtained average precisions are summarized in Table I.

TABLE I
AVERAGE PRECISION ON THE THREE INDEPENDENT TEST SUBSETS.

Method	AP 1	AP 2	AP 3
MoG single stage (validated)	98,07%	75,81%	84,07%
MoG delayed (validated)	99,86%	80,29%	49,01%
MoG two stage (validated)	99,43%	82,89%	55,67%
HoG single stage (validated)	99,77%	78,88%	91,26%
HoG two stage (validated)	99,80%	86,59%	91,84%
texture + MoG (not validated)	-	-	97,07%

Note that the approach based on texture images (texture+MoG) has been evaluated only in the hardest conditions (night-time, test subset 3) due to huge computational requirements. The obtained results are superior to all other approaches (cf. Table II) despite the fact that the decomposition has been performed with the default parameter λ [7].

TABLE II
BEST DETECTION RESULTS IN NIGHT-TIME VIDEO (TEST SUBSET 3)

Method	TP	TN	FP	FN
HoG one stage (validated)	191	2890	22	24
HoG two stage (validated)	174	2896	16	41
texture + MOG (not validated)	208	2908	4	7

The comparison of precision-recall curves is shown in Figures 3, 4 and 5. Desired operating conditions for each detection model are located close to the upper right corner in each graph. We note that the delayed and the two-stage MoG models perform very good on daylight datasets but fail during night-time. A closer examination showed that this is caused by the delaying which in some cases makes it very hard for the model to pick-up in time the illumination changes during twilight. The best performance is obtained by the HoG detection models who are therefore a clear winner of this evaluation, if we disregard texture decomposition due to computational requirements illustrated below. We note that the two-stage variants do consistently better, especially for the dataset 2. We believe that combining HoG and HoH (histogram of hue) features would actually add to current results, and they shall therefore be revisited in our future work.

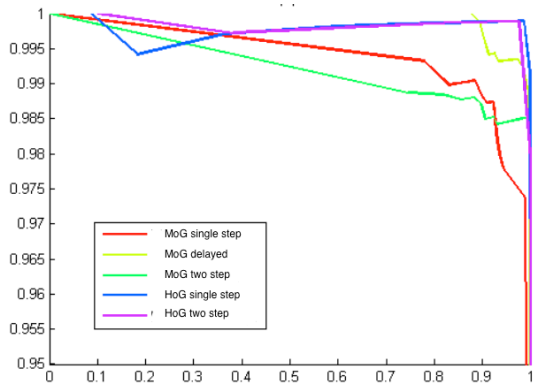

Fig. 3. Comparison of precision-recall curves on the test subset 1.

The average frame processing time (model update + detection) for the three main approaches is shown in Table III. The measurements were obtained using an Intel i5 1.7 GHz processor. The reason behind fast processing of HoG models are optimizations based on integral images.

TABLE III
AVERAGE FRAME PROCESSING TIME

Method	per-pixel MoG	per-patch HoG	per-pixel texture+MoG
Time	3 ms	1-2 ms	300-500 ms

Texture decomposition shows great potential for building background models invariant to brightness changes, but it also adds a great deal of computational complexity, which makes it inappropriate for implementations on ARM processors.

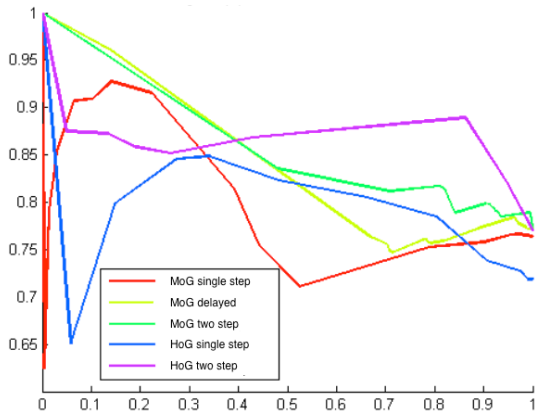

Fig. 4. Comparison of precision-recall curves on the test subset 2.

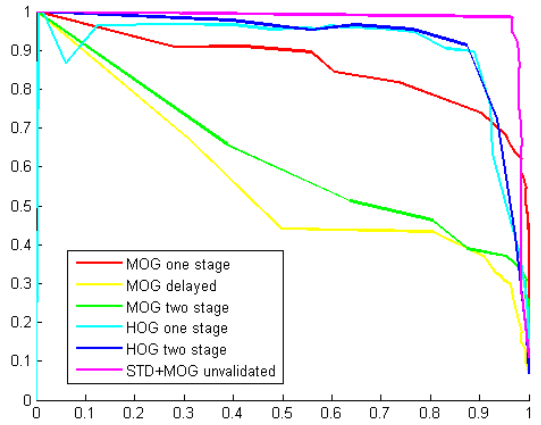

Fig. 5. Comparison of precision-recall curves on the test subset 3.

VI. Conclusion

We have addressed three practical issues which arose while applying background modelling for detecting vehicle presence in urban intersection video. Firstly, we have studied two alternative background models for decreasing sensitivity to disturbances due to illumination changes and automatic adjustments in camera hardware (mostly gain and shutter). The background model based on HOG cells has achieved nearly the same detection accuracy as texture decomposition while requiring much less computational resources. Secondly, we have shown that the HOG background model paired with the integral gradient images allows to have both the best of all worlds: the robustness to brightness-related disturbances and low computational complexity.

Thirdly, we have proposed a two-stage modification to the widely used Gaussian update approach (exponential moving average) in order to better approximate the classic running average in the desired time interval. This modification consistently improved the results of HoG models, but failed for MoG models during twilight due to fast daylight illumination changes. Further experiments shall show whether this can be improved by reducing the delay of the background model.

Future work shall be devoted to the extraction of additional features such as vehicle class, speed and inter-vehicle distance. Also we shall dedicate time to achieve combination of histogram approaches using both oriented gradients and hue values in order to try to improve presented results.

Acknowledgments

This work has been supported by the project VISTA - Computer Vision Innovations for Safe Traffic, IPA2007/HR/16IPO/001-040514 which is co- financed by the European Union from the European Regional Development Fund.

The three datasets have been kindly provided by Peek Promet d.o.o.

References

[1] O. Barnich and M. V. Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, 2011.

[2] J. Batista, P. Peixoto, C. Fern, and M. Ribeiro. A dual-stage robust vehicle detection and tracking for real-time traffic monitoring. In *Proceedings of the IEEE ITSC 2006 2006 IEEE Intelligent Transportation Systems Conference*, pages 528–535, 2006.

[3] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1937–1944, Washington, DC, USA, 2011. IEEE Computer Society.

[4] S.-C. S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. *Visual Communications and Image Processing 2004*, 5308(1):881–892, 2004.

[5] S.-C. S. Cheung and C. Kamath. Robust background subtraction with foreground validation for urban traffic video. *EURASIP J. Adv. Sig. Proc.*, 2005(14):2330–2340, 2005.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[7] J. Duran, B. Coll, and C. Sbert. Chambolle's Projection Algorithm for Total Variation Denoising. In *IPOL*, volume 3, pages 311–331, 2013.

[8] M. Harville. A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. In *ECCV (3)*, pages 543–560, 2002.

[9] S. Herrero and J. Bescós. Background subtraction techniques: Systematic evaluation and comparative analysis. In *Proceedings of the 11th International Conference on Advanced Concepts for Intelligent Vision Systems*, ACIVS, pages 33–42, Berlin, Heidelberg, 2009. Springer-Verlag.

[10] S. H. Kim, J. Shi, A. Alfarrarjeh, D. Xu, Y. Tan, and C. Shahabi. Real-time traffic video analysis using intel viewmont coprocessor. In *DNIS*, pages 150–160, 2013.

[11] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, Nov. 1992.

[12] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, pages 2246–2252, 1999.

[13] F. Tiburzi, M. Escudero, J. Bescós, and J. M. M. Sanchez. A ground truth for motion-based video-object segmentation. In *ICIP*, pages 17–20. IEEE, 2008.

[14] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):780–785, July 1997.

[15] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *ICPR (2)*, pages 28–31, 2004.