Motion Detection Using Adaptive Temporal Averaging Method

Boris NIKOLOV, Nikolay KOSTOV

Dept. of Communication Technologies, Technical University of Varna, 1 Studentska Str., 9010 Varna, Bulgaria

boris.nikolov@yahoo.com, n_kostov@mail.bg

Abstract. *Motion detection methods are widely integrated in modern intelligent video surveillance systems. Many of these methods use background subtraction techniques to separate the foreground objects from the background. Temporal averaging is one of the most commonly used and simple method for background subtraction. In this paper we propose new version of the original Temporal averaging algorithm. The speed of updating the background model has been modified to be adaptive and determined by pixel difference. Another approach with simultaneously adaptive threshold and background update speed is also proposed. Our goal is increasing the F-measure of the method by making the algorithm more versatile for different scene scenarios. Experimental results are shown and analyzed. The quality parameters of the original method and the proposed method are compared.*

Keywords

Motion detection, background subtraction, video surveillance.

1. Introduction

Motion detection is part of every modern video surveillance system. Detecting the motion is not the only function of the intelligent monitoring systems, but separating the moving objects in the foreground of the unmoving objects in the background.

Many motion detection methods have been proposed in the recent years. Background subtraction methods are most popular techniques used for this separation [1-3]. More precise and complex methods are statistical methods, like Mixture of Gaussians [4-8], Kernel Density Estimation [9], Eigenbackgrounds [10].

This paper is focused on the Temporal Averaging Method (TAM) [11-13] which is a simple method for background subtraction. In this paper the original method is shown and new variants of the algorithm are proposed.

The first step of the original TAM method is to create the background model. It represents relatively unmoving part of the scene. For each frame a new background model $B_{t(x,y)}$ is estimated by the following rule [12],

$$B_{t+1_{(x,y)}} = \alpha I_{t_{(x,y)}} + \left(1-\alpha\right) B_{t_{(x,y)}} \tag{1}$$

where $I_{t_{(x,y)}}$ is the current pixel value, t is the frame number, (x,y) is the pixel location in the image and α is learning rate.

Then the difference $D_{t(x,y)}$ between the current frame and the background is given by,

$$D_{t_{(x,y)}} = \left| I_{t_{(x,y)}} - B_{t_{(x,y)}} \right|. \tag{2}$$

The pixels whose difference value is higher than a given threshold T are classified as a foreground.

The algorithm estimates the pixels as foreground $M_{t(x,y)}$ by the rule, [12],

$$M_{t_{(x,y)}} = \begin{cases} 0, & D_{t_{(x,y)}} \le T \\ 1, & D_{t_{(x,y)}} > T \end{cases}. \tag{3}$$

2. Our Approach

The original TAM method is relatively accurate in foreground-background estimation when lighting of the scene is constant. But existence of sudden illumination changes or repeating backgrounds such as waving trees, results in high number of false positive pixels and lower levels of quality parameters. The problem with the original method is in the speed of updating the background model determined by α,(1). The method assumes that α is equal for all pixels in the current frame. This is not the best option when the background is changing very fast. So, if the speed of updating could be adaptive to each pixel difference $D_{t(x,y)}$ there would be faster algorithm reaction to sudden light changes and repeating backgrounds.

Furthermore in very noisy and dynamic background scenes, the above mentioned approach wouldn't be enough. To reduce the number of the false positives pixels in the estimated scene an adaptive threshold will be applied. In the original algorithm the threshold T is constant (3). Like in the first approach, we make the threshold T adaptive to each pixel difference $D_{t(x,y)}$.

3. The Algorithm

Variant A: The algorithm first step is the same as in the original Temporal Averaging Method. Now background $B_{t(x,y)}$ is estimated by,

$$B_{t+1_{(x,y)}} = \alpha_{t_{(x,y)}} I_{t_{(x,y)}} + \left(1-\alpha_{t_{(x,y)}}\right) B_{t_{(x,y)}} \tag{4}$$

where $\alpha_{t_{(x,y)}}$ is adaptive learning rate which will be analyzed later in the paper. Next step is to estimate the absolute difference between the current frame and the background. It is given by (2).

To modify speed of updating the background to be adaptive, the parameter $\alpha_{t_{(x,y)}}$ (4), should be adaptive for the activity in each pixel of current frame. The simplest way to determine $\alpha_{t_{(x,y)}}$ is to use the value of the difference $D_{t(x,y)}$, (2). High value of $D_{t(x,y)}$ corresponds to significant variation in the pixel value and $\alpha_{t_{(x,y)}}$ should be increased. The rule of updating $\alpha_{t_{(x,y)}}$ is similar to (1):

$$\alpha_{t+1(x,y)} = \beta.\left(\frac{C.D_{t(x,y)}}{N}\right) + \left(1-\beta\right).\alpha_{t(x,y)} \tag{5}$$

where β is learning rate of the speed of updating $\alpha_{t_{(x,y)}}$, C is an user set parameter that determines the range of $\alpha_{t_{(x,y)}}$ and N is the dynamic range of the processed signal in levels (which can be intensity or color signal). The goal is to increase the speed of updating the background when sudden changes occur in the background. For example these could be illumination changes, new unmoving objects, noise in the image or repeating backgrounds such as waving trees.

The final step is to estimate the foreground mask, $M_{t(x,y)}$ as same as (3).

Variant B: In this variant of the algorithm we make threshold T of the foreground mask $M_{t(x,y)}$, (3), adaptive for each pixel of the current frame. Pixels that represent dynamic backgrounds assume frequently high levels of difference $D_{t(x,y)}$. To prevent occurring false positive alarm the threshold $T_{t(x,y)}$ should be greater than $D_{t(x,y)}$. Updating the threshold T for a current frame t at a current pixel location (x,y) is given by the rule:

$$T_{t+1_{(x,y)}} = \frac{\alpha_{t_{(x,y)}}}{C}.\left(D_{t_{(x,y)}} + \Delta T\right) + \left(1 - \frac{\alpha_{t_{(x,y)}}}{C}\right).T_{t_{(x,y)}} \tag{6}$$

where $D_{t_{(k,j)}}$ is the current level of the difference, ΔT is constant threshold summed over the current value of $D_{t_{(k,j)}}$. The function of ΔT is increasing the threshold over the difference and reducing the number of false positives pixels.

The other part of the algorithm is same as the proposed Variant A. The background $B_{t(x,y)}$ is updated according to (4).

The absolute difference $D_{t_{(k,j)}}$ and the learning rate $\alpha_{t_{(x,y)}}$ are given by (2) and (5), respectively. The foreground mask is estimated by the modified equation (3),

$$M_{t_{(x,y)}} = \begin{cases} 0, & D_{t_{(x,y)}} \le T_{t_{(x,y)}} \\ 1, & D_{t_{(x,y)}} > T_{t_{(x,y)}} \end{cases} \tag{7}$$

where $T_{t_{(x,y)}}$ is the adaptive threshold from (6).

4. Experiments and Results

The investigated methods are implemented in Matlab. The classic Temporal Averaging Method and the variants of Adaptive Temporal Averaging Method are compared. All methods are executed for three different videos. The first processed video footage shows a group of cars moving through a city crossway. The second footage shows the same crossway at night. The third video shows people in movement and waving trees. The shooting camera is stationary. The frame rate is 25 fps and the resolution is 720x576 pixels. The lighting of the first scene is equal for all the time of the footage. The contrast of the night video is lower and there are sudden illumination changes. The waving trees in the third video footage increase the number of false positives. That prevents correct motion detection. A summary of experimental videos characteristics is given in Tab. 1.

Scene	Duration	Frame rate	Image size
Day	10s	25 fps	768x576
Night	12s	25 fps	768x576
Windy	9s	25 fps	768x576

Tab. 1. Summary of experimental videos characteristics.

The quality of background subtraction is represented as a function of each adjustable parameter in (3), (5), (6). For quantitative evaluation of the quality the *F-measure* is used [14]. It is a trade-off between parameters *recall* and *precision* and is given by,

$$F-measure = \frac{2*recall*precision}{recall+precision} \tag{8}$$

where
$$recall = \frac{TP}{TP+FN} \;, \tag{9}$$

$$precision = \frac{TP}{TP+FP} \;. \tag{10}$$

In (9) and (10) *TP* is the number of the true positives pixels which are correctly classified foreground pixels. *FP* is a number of false positive pixels. These pixels are background pixels, wrongly classified as a foreground. *FN* is a number of false negative pixels, which are foreground pixels, wrongly segmented as background.

In the beginning of the experiments the original method and Version A of the proposed method are compared. The results of estimating three different videos are analyzed. The original video frame is compared to foreground frame. The *F-measure* is calculated when the foreground image is subtracted to a ground truth image. In Fig. 1 (a) original video frames of the three processed scenes, (b) the foreground images obtained after processing the video by Variant A of the proposed algorithm, (c) ground truth images are shown. There are some obvious challenges in the backgrounds of the scenes: some noise in the day video, low contrast and flashing lamps in the night video and large waving threes in the windy video footage.

(a) *(b)* *(c)*

Fig. 1. (a) Original video frames, (b) Foreground frames; (c) Ground truth.

The graph in Fig. 2 shows the value of the *F-measure* parameter [14] as a function of the threshold *T* (3). The original Temporal Averaging Method and the proposed method are compared by executing the first video footage. The adaptive method is more accurate by 1% than the original. This is minor improvement because in daylight the footage doesn't represent any specific conditions like dynamic backgrounds or sudden illumination changes.

In the next graph in Fig. 3 the results of processing the night video are shown. Again, the new method is more correct than the original one. The maximum improvement in *F-measure* is 4%. In this video there are non-constant lighting conditions and low contrast. This advance is determined by the adaptive learning rate $\alpha_{t_{(x,y)}}$ (5) and the

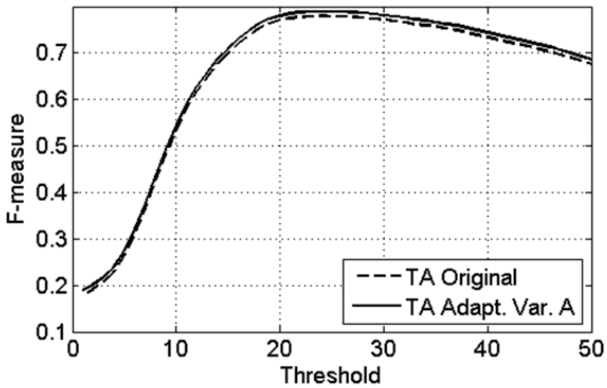


Fig. 2. *F-measure* as a function of the threshold *T* (daylight footage).

faster speed of updating the noisy background. The result is reducing the false positive pixels. When the traffic lights turn green the Adaptive Temporal Averaging Method updates the change in the background faster than the original method. This results in increasing precision and increasing the *F-measure*.

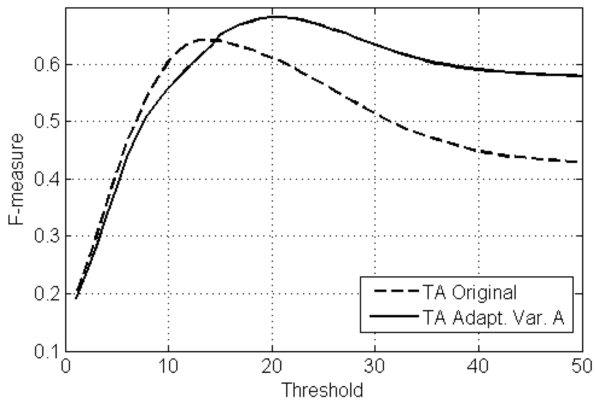


Fig. 3. *F-measure* as a function of the threshold *T* (night footage).

Another challenge is posed in the third scene. The footage is captured in very windy weather conditions and there are waving trees. The proposed algorithm is more effective again. An increase of 4% is obtained in Fig. 4.

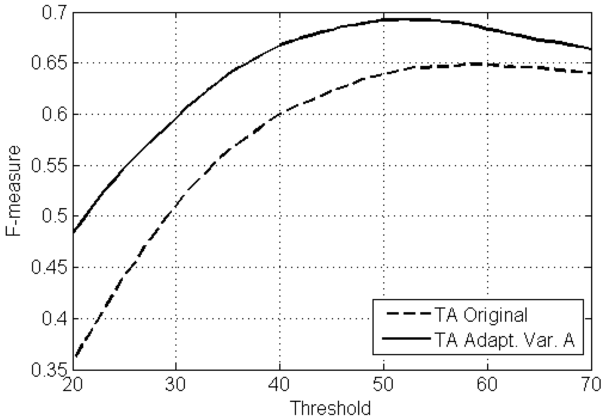
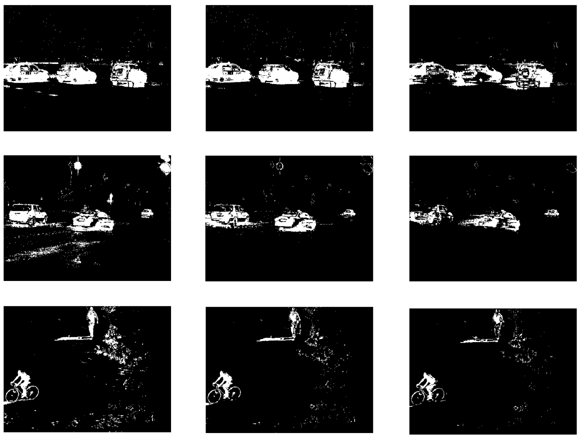


Fig. 4. *F-measure* as a function of the threshold *T* (windy footage).

The experimental results obtained by studying the learning rate β and the parameter C, are interesting. The C parameter, (5), determines the range of α. The best set of its value depends on the speed of the moving objects. High speed of moving corresponds to high levels of α. For example, α usually varies from 0.1 to 0.001. In Fig. 5 the results of foreground detection when C is executed for three different videos are shown. In Fig. 5a, foreground images processed when C is too low are shown. In Fig. 5b, the detected image of highest accuracy is shown and the image in Fig. 5c corresponds to very high level of C. The most successful results are obtained when C is in range of 0.2 to 0.4.



(a) (b) (c)

Fig. 5. (a) Motion detection by low levels of C; (b) Motion detection by correct levels of C; (c) Motion detection by high levels of C.

The experiment results when β (5) is a variable are shown in Fig. 6a and Fig. 6b. Its range is $0 < \beta \leq 1$. This parameter determines the speed of updating α. As β is increasing the noise resistance of the algorithm will be higher and more false positives pixels in the background will be eliminated. But high levels of β such as 0.1 and more may cause problems in detection of slow moving objects in the foreground. They are classified as a background so fast and the result is a trace of false positive pixels after them. This is illustrated in Fig. 7.

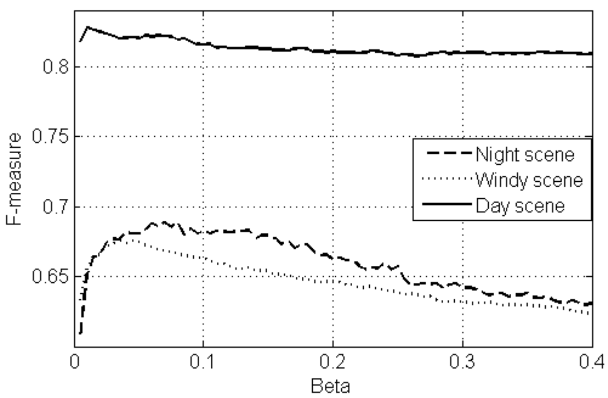


Fig. 6a. *F-measure* as a function of learning rate β – Variant A.

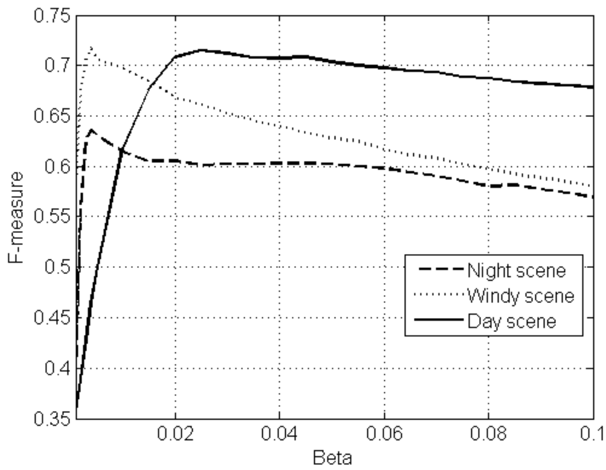


Fig. 6b. *F-measure* as a function of learning rate β – Variant B.

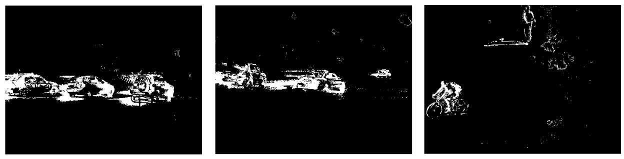


Fig. 7. Trace of false positive pixels after slow moving objects.

The second variant of our adaptive algorithm is experimentally examined, (6), (7). The results of executing the method for the three different videos are shown. The adaptive threshold value is high for the moving objects and the number of the false positive pixels is low. As a result the precision, (10), increases, but the recall, (9), of the algorithm decreases. *F-measure* is a trade-off between these two parameters and the final effect of the algorithm efficiency depends on the conditions of the estimated scene.

The performance of Variant B of the proposed algorithm is compared to other methods by ROC characteristics (Receiver Operating Characteristics). This is shown in Fig. 8, 9, 10 for three different videos.

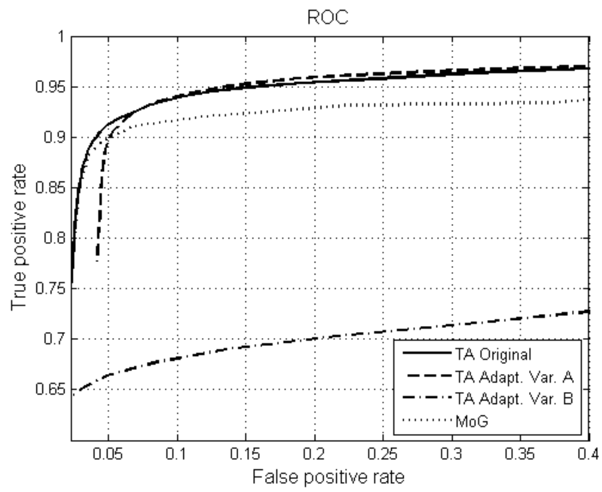


Fig. 8. ROC of TA Original, TA Adapt Var. A, B, MoG (daylight footage).

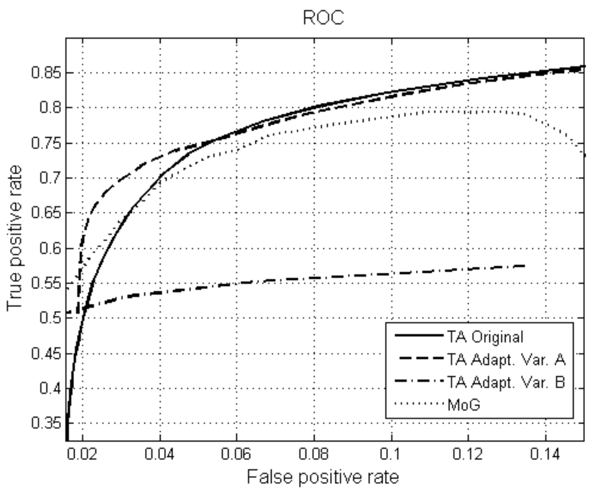


Fig. 9. ROC of TA Original, TA Adapt Var. A, B, MoG (night footage).

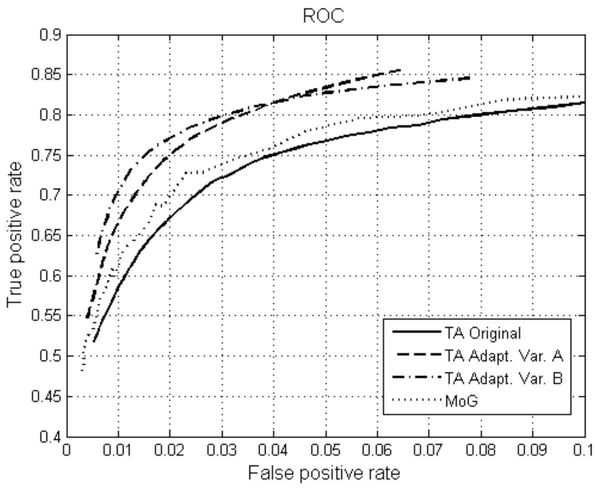


Fig. 10. ROC of TA Original, TA Adapt Var. A, B, MoG (windy footage).

It is obvious that there is no positive effect of applying the second variant of the algorithm in the first and in the second scene. But in the third scene there are large waving trees and relatively small moving objects. Variant B of the algorithm provides more than 7% increase in the *F-measure*. The adaptive threshold *ΔT* prevents false positives to penetrate in the foreground image. In Fig. 11 the original images and visualization of the adaptive threshold level *T* are shown. Bright areas in the frame represent pixels with high level of the threshold *T*.

A summary of the *F-measure* highest values is shown in Tab. 2. The presented methods are compared to each other. There is obvious advance of the Adaptive Temporal Averaging Method. Variant A is suitable for general types of scenes, day or night. Variant B obtains advance only in scenes with dynamic and noisy background. In the bottom of Tab. 2 the results of motion detection by Mixture of Gaussian background subtraction method (MoG), [1] are shown. Despite this complex method provides minor advance in *F-measure* level, it is not always the best option for intelligent video surveillance systems. Real time implementation of MoG method requires large computational power. The best choice of motion detection method is a trade-off between quality of detection and simplicity in processing the video signals. Therefore when available computational power is limited, our algorithm is more appropriate than other complex methods.

a) *b)*

Fig. 11. a) Original image; b) Threshold level.

Method	*Parameter*	*Scene*		
		Day	Night	Windy
TAM	**F-measure**	**0.78**	**0.64**	**0.65**
	Precision	0.77	0.59	0.7
	Recall	0.79	0.7	0.6
Adaptive TAM Var. A	**F-measure**	**0.79**	**0.68**	**0.69**
	Precision	0.7	0.7	0.73
	Recall	0.89	0.67	0.66
Adaptive TAM Var. B	**F-measure**	**0.72**	**0.63**	**0.72**
	Precision	0.82	0.82	0.76
	Recall	0.64	0.51	0.68
MoG	**F-measure**	**0.81**	**0.76**	**0.68**
	Precision	0.75	0.8	0.77
	Recall	0.88	0.72	0.61

Tab. 2. Summary of the F-measure the highest values,

	Variant A			Variant B		
	β	*C*	*T*	*β*	*C*	*ΔT*
Daylight video	0.02	0.2	31	0.025	0.2	12
Night video	0.06	0.2	21	0.005	0.2	7
Windy video	0.035	0.3	50	0.005	0.3	35
Night video 2	0.06	0.4	29	0.007	0.2	18
Black and white video	0.04	0.3	21	0.006	0.2	22

Tab. 3. Summary of the most successful values of the algorithm parameters.

The results of the experiments show that the maximum performance of the method depends on the proper set of its parameters. A summary of the most successful values of the proposed algorithm parameters is shown in Tab. 3. There is extra data for more videos to help assessing the robustness of the parameters.

5. Discussion

Both proposed variants of the method have to face some challenges. Slowly moving objects could be wrongly classifies as a background. This depends on the speed of updating the background and specifically α.

The range of α is determined by the user set parameter C, (5). A proper range of α has to be set to prevent wrongly classifying the moving objects as changes in the background. According to Tab. 3 the most successful values of C occupy range of 0.2-0.4. This is quite narrow interval and the average value of 0.3 for C is appropriate for general types of scenes. In case there is heavy traffic scene, where the cars are moving in short distance α assume constantly high levels and this results in dominance of false negative pixels in the real moving objects. The learning rate β varies from 0.02 to 0.06 for Variant A and from 0.005 to 0.025 for Variant B. Scenes with sudden or gradual illumination changes or dynamic background require high values of β. The most proper value of the threshold T depends on the scene contrast and the existence of dynamic background. It is also determined by a trade-off between precision and recall.

The proposed algorithm is designed to adapt not only to sudden illumination changes. Very slowly changes in the background are also updated. For example if there is a very slowly increasing or decreasing in the lightning the minimum possible value of the difference, (2), is 1. Then, according to Tab. 3 and equation (5) the minimum possible value of α is 0.001. This learning rate is fast enough to update very slowly movements or changes in illumination.

6. Conclusion

In this paper a new method for background subtraction called Adaptive Temporal Averaging Method was proposed. Three videos representing different types of scenes were captured to test the method. The experimental results were shown. Two variants of the proposed method and the original Temporal Averaging Method were compared.

The Adaptive Temporal Averaging Method compared to the original method achieves increasing in *F-parameter*. The new algorithm advantage depends on the conditions of the scene. Variant A of the algorithm is more accurate than Variant B and original Temporal Averaging Method when the background image is stable and the moving objects are large. Variant B of the proposed algorithm is more appropriate to implement when dynamic backgrounds are frequently present in the scene. Also, in the original TAM method there exists a trace of false positive pixels appeared after the moving object. Variant B of the proposed method prevents this disadvantage. The reaction of lighting condition variations is faster enough, so it is suitable for estimating night video footages or not constant lighting scenes.

The complexity of the algorithm is minor and the required computational power can be maintained by not expensive digital signal processors.

Acknowledgement

The carried out research is funded by the project BG051PO001-3.3.06-0005, Program ‘Human Resources Development’, Technical University of Varna, Bulgaria.

References

[1] VARGAS, M., TORAL, S. L., BARRERO, F., MILLA, J. M. An enhanced background estimation algorithm for vehicle detection in urban traffic video. In *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*. Beijing (China), Oct 2008, p. 784–790.

[2] MANZANERA, A., RICHEFEU, J. C. A new motion detection algorithm based on sigma-delta background estimation. *Pattern Recognition Letters In Advances in Visual Information Processing*, Feb 2007, vol. 28, no. 3, p. 320–328.

[3] MCFARLANE, N. J. B., SCHOFIELD, C. P. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, May 1995, vol. 8, no. 3, p.187–193.

[4] STAUFFER, C., GRIMSON, W. E. L. Adaptive background mixture models for real-time tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, June 1999, vol. 2, p. 246–252.

[5] BOUWMANS, T., EL BAF, F., VACHON, B. Background modeling using mixture of Gaussians for foreground detection - a survey. *Recent Patents on Computer Science*, November 2008, vol. 1, no. 3, p. 219–237.

[6] DALLEY, G., MIGDAL, J., GRIMSON, W. Background subtraction for temporally irregular dynamic textures. In *WACV 2008*. Colorado (USA), January 2008, p. 1–7.

[7] DICKINSON, P., HUNTER, A. Scene modelling using an adaptive mixture of Gaussians in colour and space. In *IEEE Conf. on Advanced Video and Signal based Surveillance*, *AVSS 2005*. Como (Italy), September, 2005, p. 64–69.

[8] WANG, H., MILLER, P. Regularized online mixture of Gaussians for background subtraction. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance, AVSS 2011*. Klagenfurt (Austria), September, 2011, p. 249–254.

[9] ELGAMMAL, A., DURAISWAMI, R., HARWOOD, D., DAVIS, L. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE,* July 2002, vol. 90, no. 7, p. 1151–1163.

[10] HALL, P., MARSHALL, D., MARTIN, R. Merging and splitting eigenspace models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000, vol. 22, no. 9, p. 1042–1049.

[11] HEIKKILA, J., SILVEN, O. A real-time system for monitoring of cyclists and pedestrians. In *Proceedings of the Second IEEE Workshop on Visual Surveillance, (VS'99).* July 1999, p. 74–81.

[12] FRIEDMAN, N., RUSSELL, S. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the*

Thirteenth Conference (UAI-1997). San Francisco (USA), 1997, p. 175–181.

[13] HA, D. M., LEE, J. M., KIM, Y. D. Neural-edge-based vehicle detection and traffic parameter extraction. *Image and Vision Computing,* 2004, vol. 22, no. 11, p. 899–907.

[14] AGARWAL, S., AWAN, A., ROTH, D. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2004, vol. 26, no. 11, p. 1475–1490.

About Authors ...

Boris NIKOLOV was born in Varna, Bulgaria, on July 6, 1984. He received Dipl. Eng. degrees from the Technical University of Varna in 2009. Since 2009 he is a PhD student at the Communication Technologies Department, Technical University of Varna, Bulgaria. Since 2010 he joined the same department as an Assistant Professor. His research interests include image processing, video surveillance, video compression technologies and digital television.

Nikolay KOSTOV is an Associate Professor of Communication Science at the Technical University of Varna, Bulgaria. He received his Dipl. Eng. and PhD from the National Military University of Veliko Turnovo, Bulgaria, in 1993 and 1999, respectively. His research interests include source/channel coding, modulation techniques, communications over fading channels, and digital video and audio processing.