Investigation of Mixture of Gaussians Method for Background Subtraction in Traffic Surveillance

Boris Nikolov[1], Nikolay Kostov[2] , Slava Yordanova[3]

***Abstract* – Many background subtraction techniques have been developed in the past years to improve the precision of motion detection in video surveillance systems. Separating the moving objects from the background is a goal in every modern video surveillance system. Mixture of Gaussians (MoG) is one of the most complex method used for motion detection in video sequences.**

This paper further investigates the MoG method. The algorithm is implemented in Matlab and a typical traffic video is estimated. The accuracy of the algorithm is measured as a function of each variable parameter. An optimal set of parameters along with a filter are proposed in order to increase the performance.

***Keywords* – Mixture of Gaussians, motion detection, background subtraction, video surveillance.**

I. Introduction

The idea of the background subtraction is to separate the foreground moving objects from the background objects in the scene. Mixture of Gaussians (MoG) is a complex method for background subtraction [1]-[10]. There are many parameters to be set. This paper continues our research [2] and investigates the relation between algorithm parameters and the precision of the background subtraction. The results of using median filter after image processing are given.

II. Mixture of Gaussians method

The MoG method describes each pixel in the frame by using multiple Gaussian distributions. Each pixel is represented by a distribution with its associated variance, weight and mean.

The probability of observing the current pixel value *x* at time *t* at a particular pixel location is given by [1]

$$P\left(x\right)=\sum_{i=1}^{K}\omega_{i,t}\eta(x;\mu_{i,t},\Sigma_{i,t}), \qquad (1)$$

[1]Boris Nikolov is with the Technical University of Varna, Faculty of Electronics, Department of Communication Technologies, Bulgaria, Varna 9010, 1 Studentska Str.
E-mail: boris.nikolov@yahoo.com

[2]Nikolay Kostov is with the Technical University of Varna, Faculty of Electronics, Department of Communication Technologies, Bulgaria, Varna 9010, 1 Studentska Str.
E-mail: n_kostov@mail.bg

[2]Slava Yordanova is with the Technical University of Varna, Faculty of Computing and Automation, Department of Computer Science and Engineering, Bulgaria, Varna 9010, 1 Studentska Str.
E-mail: slava_y@abv.bg

where K is the number of Gaussians distributions representing each pixel, $\omega_{i,t}$ is the weight of the i^{th} Gaussian at time t, η is the Gaussian probability density function with parameters: x is the current pixel, $\mu_{i,t}$ is the mean of the i^{th} distribution at time t, and $\Sigma_{i,t}$ is the covariance of the i^{th} distribution at time t.

The Gaussian probability density function η is given by [1]

$$\eta\left(x;\mu,\Sigma\right)=\frac{1}{\left(2\pi\right)^{\frac{n}{2}}\left|\Sigma\right|^{\frac{1}{2}}}e^{-\frac{1}{2}\left(x-\mu\right)^{T}\Sigma^{-1}\left(x-\mu\right)}. \qquad (2)$$

The covariance matrix is formed by [1],

$$\Sigma_{k,t}=\sigma_{k}^{2}I\ , \qquad (3)$$

where I is the image sequence.

The mean and the dispersion are updated by the rule [1]

$$\mu_{t}=\left(1-\rho\right)\mu_{t-1}+\rho X_{t}\,, \qquad (4)$$

$$\sigma_{t}^{2}=\left(1-\rho\right)\sigma_{t-1}^{2}+\rho\left(X_{t}-\mu_{t}\right)^{T}\left(X_{t}-\mu_{t}\right), \qquad (5)$$

where ρ is learning rate determined by

$$\rho=\alpha\eta\left(X_{t}\mid\mu_{k},\sigma_{k}\right). \qquad (6)$$

A particular value *x* being observed at a pixel location has a high probability if it is close to the mean of Gaussian distribution with a high weight and a low variance. So, this is the Gaussian distributions that best describes each pixel. Every new pixel value is checked again if there is a match with the existing K Gaussian distributions. By default, a match is defined as a pixel value within 2.5 standard deviations of a distribution [1]. In this paper the deviation threshold will be marked as D.

The weight of the i^{th} Gaussian at time *t* is given by [1]

$$\omega_{i,t}=\left(1-\alpha\right)\omega_{i,t-1}+\alpha\left(M_{i,t}\right), \qquad (7)$$

where α is a learning rate, and $M_{i,t}$ is 1 for the model which matched and 0 for the remaining models.

In case there is no match a new Gaussian distribution is created with a mean equal to the current pixel value. The new distribution replaces the distribution with the lowest weight and highest variance. It is assumed that the background is represented by Gaussian distributions with the highest weight and lowest variance. To estimate the background, the distributions are first sorted in order of decreasing ω/σ. The pixels that belong to the background are the first B distributions and B is given by

$$B = \arg\min_{c}\left(\sum_{k=1}^{c}\omega_k > T\right), \tag{8}$$

where T is threshold which separates the background from the foreground.

III. Experiments and Results

The investigated method is implemented in Matlab. The processed video footage shows a car passing by the street. The shooting camera is stationary. The lighting of the scene is equal for all the time of the scene. The speed of the car is 22 km/h. The frame rate is 15fps and the resolution is 320x240 pixels.

The performance of the MoG algorithm depends on appropriate set of the parameters: the threshold T (Eq. (11)), the learning rate α (Eq. (7)), the deviation threshold D and the number of Gaussians distributions K. The algorithm is executed for different values of the parameters.

The accuracy is represented as function of each adjustable parameter. For quantitative evaluation of the accuracy the *F-measure* is used [3]. The above mentioned *F-measure* is a trade-off between parameters *recall* and *precision* and is defined by

$$F-measure = \frac{2 * recall * precision}{recall + precision}, \tag{9}$$

where *recall* and *precision* are given by

$$recall = \frac{TP}{TP+FN}, \tag{10}$$

$$precision = \frac{TP}{TP+FP}. \tag{11}$$

In (10) and (11) TP is the number of the true positives pixels which are correctly classified foreground pixels and FP is the number of the false positives pixels. These pixels are background pixels, wrongly classified as a foreground. FN is the number of the false negative pixels, which are foreground pixels, wrongly segmented as background.

To prevent the noise in the foreground image a 3 by 3 two dimensional median filter is applied. The results of filtering the image are compared to the original estimated foreground image.

A. Threshold

The MoG method is executed for 12 values of the parameter T. The graph in Fig. 1 shows the accuracy parameter *F-measure* as a function of the threshold T. The lower curve represents the result of the experiment without filtering. The upper curve shows the value of the *F-measure* after processing the foreground image with the median filter. In Fig. 2 are shown the original frame, the estimated foreground image and the foreground image after filtering. The figure represents the results for three values of the threshold T, respectively, too low (Fig. 2a), highest *F-measure* (Fig. 2b) and too high (Fig. 2c).

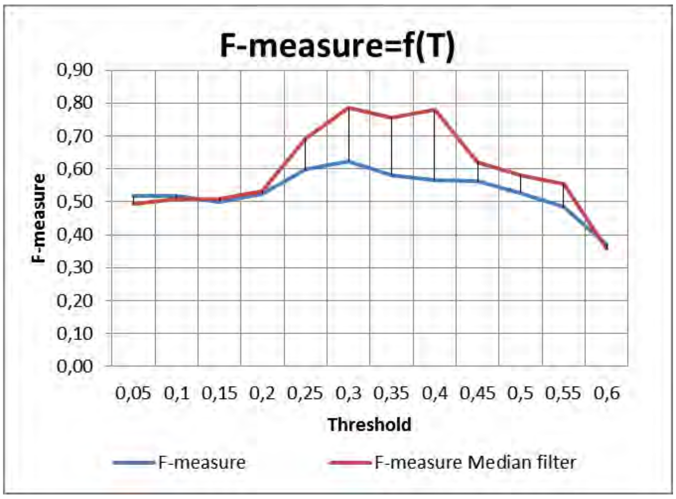

Fig. 1. *F-measure* as a function of the threshold T.

Fig. 2. Original image, foreground image and filtered foreground image executed for three values of T, $T=0.05$ (a); $T=0.3$ (b); $T=0.55$ (c)

B. Deviation threshold

The method accuracy is tested by twelve values of deviation threshold D which determines the match between the new pixel value and the current distribution. In Fig. 3 the experimental results are shown. The highest value of the *F-measure* is 0.69 and is received when $D=4$, contrary to [1]. The upper graph represents the results of filtering the foreground image with median filter. Obtaining the highest performance after median filtering at $D=2.5$ is an interesting result. The *F-measure* value is increased with 16%. The highest difference in accuracy between the original foreground image and filtered foreground image is at $D = 1.5$ when an increasing of 31% of the *F-measure* is obtained. In Fig. 4 is shown the estimated foreground image for four different values of the deviation threshold D, respectively original foreground image (second column) and filtered with median filter image (third column).

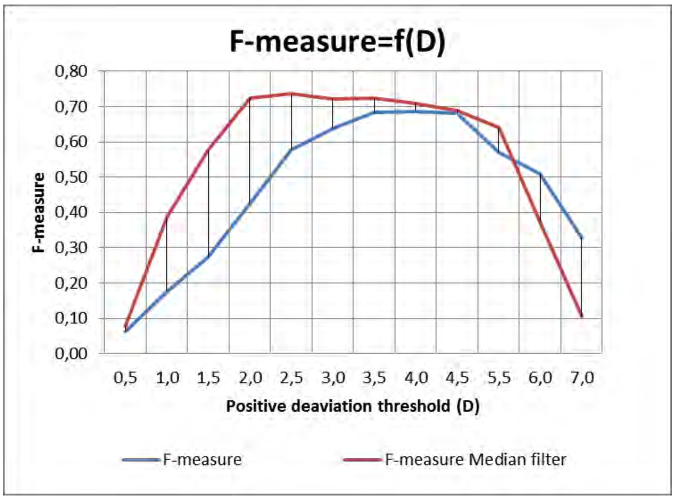

Fig. 3. *F-measure* as a function of the deviation threshold D.

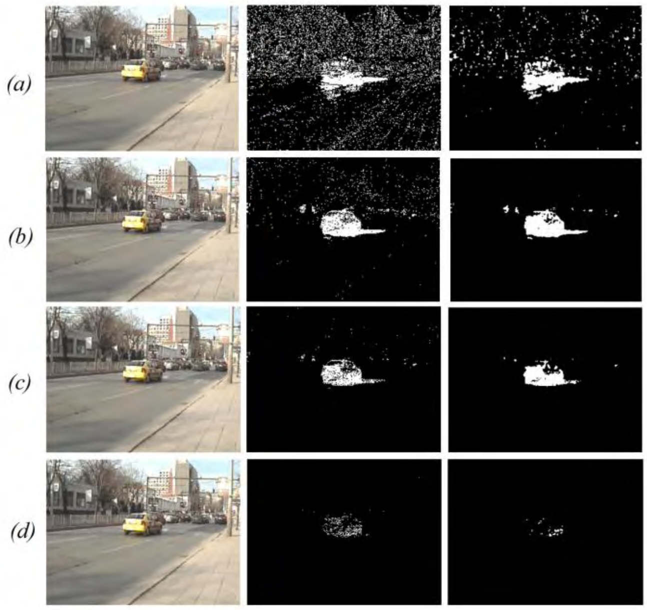

Fig. 4. Original image, foreground image and filtered foreground image executed for four values of D, $D = 1$ (a); $D = 2.5$ (b); $D = 4$ (c); $D = 7$ (d)

C. Learning rate α

The *F-measure* of the algorithm is tested for 12 different values of the learning rate α. The results are shown in Fig. 5. The accuracy remains constant from $\alpha = 0.005$ to $\alpha = 0.05$. The upper graph shows the results of filtering the foreground image with the median filter. If α is below 0.005 to many pixel will have high weight and will be classified as a background and if α is over 0.05 the weight of the unmatched pixels will easily decrease and there will be more foreground objects. In Fig. 6 are shown three cases of running the method: Fig. 6(a) $\alpha = 0.0025$, Fig. 6(b) $\alpha = 0.01$ and Fig. 6(c) $\alpha = 0.5$.

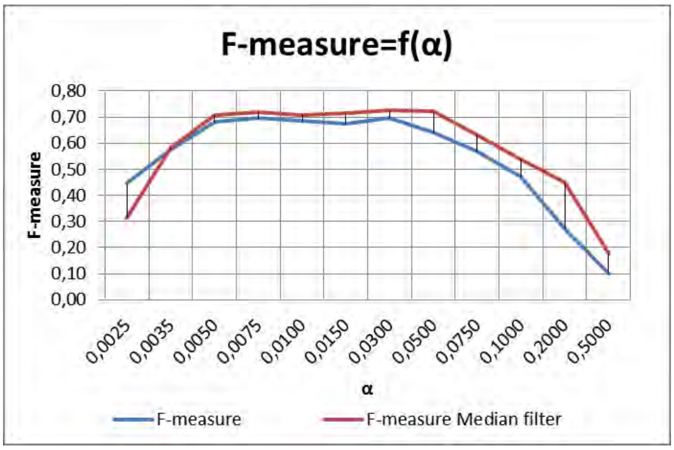

Fig. 5. *F-measure* as a function of the learning rate α.

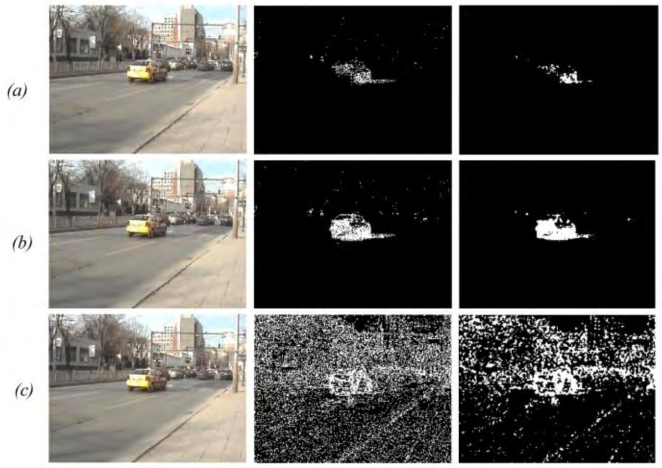

Fig. 6. Original image, foreground image and filtered foreground image executed for three values of α, $\alpha = 0.0025$ (a); $\alpha = 0.01$ (b); $\alpha = 0.5$ (c)

D. Number of Gaussians distributions K

The result of executing the algorithm for six different K Gaussian distributions is shown in Fig. 7. Increasing the number of components in MoG does not help in improving the quality of the extracted foreground image. Highest *F-measure* is obtained at $K = 3$ and $K = 4$. This is because although more components can model more distributions, simple scenes are often not characterized by complex changes, and updating components of the model causes more noise. In more complex scenes the number of Gaussian distributions should be greater than $K = 4$. For instance, with many small moving objects in low contrast and windy scenes, where are waving tress and not constantly lighting. Another problem of increasing the number of distributions is that the processing time gets longer.

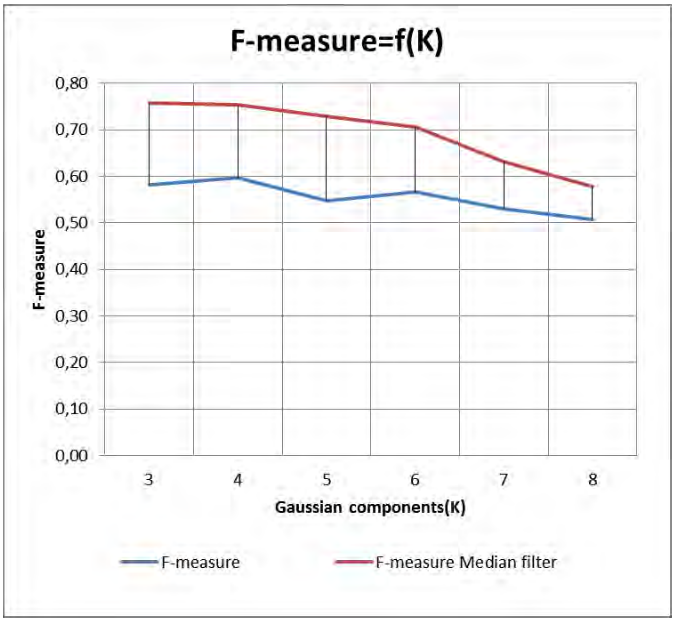

Fig. 7. *F-measure* as a function of number of the Gaussians distributions K.

IV. CONCLUSION

Background subtraction using MoG method was investigated. The results of running the algorithm and estimating the accuracy as a function of the algorithm variables were shown.

The processed video is a typical traffic surveillance footage. The value of the threshold T which determines the highest *F-Measure* of 0.62 is $T=3$ and after median filtering the *F-Measure* increases to 0.78. The highest algorithm performance depends on whether or not a median filter is used, when adjusted variable is the deviation threshold. Without filtering the highest value of *F-measure* is reached when $D=4$ and $F-measure=0.69$. Applying the filter shifts the best deviation threshold to $D=2.5$ when $F-measure=0.74$. Varying the learning rate α from $\alpha=0.005$ to $\alpha=0.05$ does not result in a significant change of accuracy. Increasing the number of Gaussian distributions K do not result in high *F-measure* but leads to a long processing time. Filtering the noise in the processed foreground image with a *3 by 3* median filter increases the accuracy of the background subtraction. The average increasing of *F-measure* is 7.8%. Further improvement should be expected with more complex filters.

The MoG background subtraction method is high performance method if the algorithm variables are adjusted correctly. The best MoG parameter set depends on specific conditions of the sequence. The algorithm is very versatile to different applications. Real time implementation of the algorithm via digital signal processor is our aim for feature experiments and research.

ACKNOWLEDGEMENT

This paper is funded by National Ministry of Education and Science of Bulgaria under project “Investigation of Communication Systems via Digital Signal Processors”, Technical University of Varna, Bulgaria, 2012.

REFERENCES

[1] C. Stauffer and W.E.L Grimson, “Adaptive background mixture models for real-time tracking,” *IEEE International Conference on Computer Vision and Pattern Recognition*, vol.2, pp. 246–252, June,1999.

[2] B. Nikolov, N. Kostov “Precision of Some Motion Detection Methods Using Background Subtraction in Traffic Surveillance Video”, *In Proceedings of ICEST Conference 2011*, Nis, Serbia, pp. 19-23, June 2011.

[3] Shivani Agarwal, Aatif Awan, and Dan Roth, “ Learning to detect objects in images via a sparse, part-based representation”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11), pp. 1475–1490, 2004.

[4] Atev S, Masoud O, Papanikolopoulos N. Practical mixtures of gaussians with brightness monitoring. *IEEE Conf on Intt Transportation Systems, Proceedings (ITS 2004)*, pp. 423-428, 2004.

[5] Dickinson P, Hunter A. Scene modeling using an adaptive mixture of Gaussians in color and space. *IEEE Conf on Advanced Video and Signal based Surveillance (AVSS 2005)*, Como, Italy, pp. 64-69, September 2005.

[6] Zang Q, Klette R. “Parameter analysis for Mixture of Gaussians”, CITR Technical Report 188, Auckland University, 2006.

[7] Shimada A, Arita D, Taniguchi R. “Dynamic control of adaptive mixture-of-gaussians background model”, *AVSS 2006*, Sydney, Australia, pp. 5 – 5, November 2006.

[8] Utasi A, Czúni L. “Reducing the foreground aperture problem in mixture of Gaussians based motion detection”, *6th EURASIP Conf Focused on Speech and Image Processing, Multimedia Communications and Services EC-SIPMCS 2007*, Maribor, Slovenia, pp. 157 – 160, 2007.

[9] El Baf F, Bouwmans T, Vachon B. “Type-2 fuzzy mixture of Gaussians model: Application to background modeling”, *International Symposium on Visual Computing (ISVC 2008)*, Las Vegas, USA, pp. 772-781, December 2008.

[10] Cuevas C, Salgado L, Garcia N. “A new strategy based on adaptive mixture of Gaussians for real-time moving objects segmentation”, *In Proc. of SPIE-IS&T Electronic Imaging, SPIE*, Vol. 6811, pp. 681111, 2008.