A QR Decomposition based Mixture Model Algorithm for Background Modeling

Mahmood Amintoosi*‡, Farzam Farbiz*†, Mahmood Fathy*
*Computer Engineering Department, Iran University of Science and Technology, Iran
†A*STAR Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
‡Mathematics Department, Tarbiat Moallem University of Sabzevar, Iran

***Abstract*— This paper presents a new algorithm for background modeling in a sequence of images, even if there are foreground objects in each frame. We develop a QR decomposition based algorithm to remove foreground pixels from the image and then we construct the background model using Mixture of Gaussian algorithm, MoG. We split the image into small blocks and construct the background blocks using R-values taken from QR decomposition which indicate the degree of significance of the decomposed parts. The simulation results show the better performance of the proposed algorithm in compare with conventional methods on modeling static background images.**

***Keywords*— Image processing, Matrix decomposition, Linear algebra, Image segmentation, Object detection.**

I. Introduction

Background segmentation is one of the most fundamental tasks in computer vision with a wide spectrum of applications from compression to scene understanding, especially for detecting moving objects in videos taken from static cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called the "background image" or "background model". There have been several background subtraction methods in the literatures [1, 2, 3, 4, 5, 6]. All of these methods try to effectively estimate the background model from the temporal sequence of the frames.

In a completely static scene, the intensity value of a pixel can be reasonably modeled by a Normal distribution . This is the underlying model of many background subtraction techniques. In the surveillance system of Stauffer and Grimson [1], which has become a standard formulation for the mixture approach in the field, an online Expectation Maximization, EM, approximation based on recursive filter was used to train the mixture background model. In [2], a single Gaussian model is used and the parameters are updated by alpha blending. Unfortunately, these approaches fail when the distribution of background pixel intensity does not fit into a single model. Mixture models were proposed to handle the backgrounds that exhibit multimodal characteristics. Porikli and Tuzel [5] modeled each pixel as a set of layered normal distributions. Lee [3] proposed a Gaussian mixture learning method, which produced good estimate of the background; even there was foreground objects in every moment.

The background segmentation involves a binary classification problem based on $P(B|x)$, where x is the pixel value at time t, and B represents the background class. With an explicit representation of the temporal distribution $P(x)$ as a mixture [3], the posterior probability can be expressed in terms of the mixture components $P(G_k)$ and $P(x|G_k)$

$$P(x) = \sum_{k=1}^{K} P(G_k)P(x|G_k) = \sum_{k=1}^{K} \omega_k . g(x; \mu_k, \sigma_k) \tag{1}$$

$$\begin{aligned} P(B|x) &= \sum_{k=1}^{K} P(B|G_k)P(G_k|x) \\ &= \frac{\sum_{k=1}^{K} P(x|G_k)P(G_k)P(B|G_k)}{\sum_{k=1}^{K} P(x|G_k)P(G_k)} \end{aligned} \tag{2}$$

One of the concentrating aspects of background mixture model in the literatures is on estimation of $P(B|G_k)$ in the above equation to distinguish which kernel belongs to background and which to the foreground. Some papers use a training phase, where the scene is empty. Others try to make decision heuristically. In [1], $P(B|G_k)$ equals to 1 for Gaussians with the highest ω/σ covering a certain percentage of observations, and 0 for all others. Lee [3] trained a sigmoid function on ω/σ to approximate $P(B|G_k)$. In [7] the Gaussians are manually labeled and remain fixed; the darkest component is labeled as shadow, the one with the largest variance labeled as vehicle and the remaining one as road.

In [8] we presented a background modeling algorithm based on the assumption that each small block in the image would reveal the background for at least a short interval of the sequence, and this short interval must be longer than any foreground interval. This is a usual case in a road situation. The key idea of the proposed method lied in the identification of the background based on these blocks using QR decomposition technique, a known method in Linear Algebra. R-values produced with QR decomposition can be applied to decompose a given system and indicate the degree of the significance of the decomposed parts. Selection of background blocks was conceptually obtained by choosing those parts which have weak contribution, according to the assigned R-values. The background model was constructed then using these selected blocks.

In this paper we improve our previous algorithm [8] in two ways i) to develop an online version of the background modeling algorithm, and ii) to consider mixture of Gaussian in background modeling. These two modifications not only improve the modeling performance, but also improve the processing speed.

The reminder of this paper is organized as follows: Section 2 explains the proposed method. Section 3 provides simulation results and Section 4 describes the conclusion and future works.

II. The Proposed Method

In this section we first briefly discuss our QR decomposition algorithm for background modeling [8]. Then we propose its online version and the hybrid algorithm of QR decomposition background modeling and the MoG method.

A. QR Decomposition based Background Modeling

The QR decomposition of P is given by $P\Pi = QR$, where $\Pi \in R^{M\times M}$ is a permutation matrix, $Q \in R^{M\times M}$ has orthonormal columns, and $R \in R^{M\times M}$ is upper triangular. The QR decomposition is uniquely determined by the permutation matrix Π, and many techniques have been proposed to compute it. The diagonal elements of matrix R are called R-values. R-values are in decreasing order and they tend to track the singular values of P [9]. The singular value decomposition has been used by many people for rule base reduction [9,10]. The key idea of using the singular value decomposition (SVD) in complexity reduction is that the singular values can be applied to decompose a given system and indicate the degree of the significance of the decomposed parts. Reduction is conceptually obtained by the truncation of those parts which have weak contribution to the output according to the assigned singular values [10]. We use this feature to distinguish background pixels from foreground ones.

In our proposed method, we split the input gray scaled image frame into M small blocks and apply QR decomposition method on each block to identify the background part. We consider the values of a particular pixel over time as a "pixel process". It is a time series of intensity values X of a particular pixel (x_i, y_i) at b^{th} block $(b = 1, \ldots, M)$, by time t :

$$X = \{X^b_{i,1}, X^b_{i,2}, \cdots, X^b_{i,t}\} \tag{3}$$

We construct matrix A^b for b^{th} block as:

$$A^b = \begin{bmatrix} X^b_{1,1} & X^b_{1,2} & \cdots & X^b_{1,t} \\ X^b_{2,1} & X^b_{2,2} & \cdots & X^b_{2,t} \\ \vdots & \vdots & \ddots & \vdots \\ X^b_{N,1} & X^b_{N,2} & \ldots & X^b_{N,t} \end{bmatrix} \tag{4}$$

where N is the number of pixels in each block. Because the proposed method is the same for all blocks, in the rest of this section we name A^b as A. Each R-value taken from QR decomposition of matrix A, is related to one of the columns of A. Since those columns of A containing only background data are almost similar to each other, the R-values corresponding to these columns will be smaller than those containing foreground objects. If we define the series Y as a sorted list of X according to the R-values, for the i^{th} pixel's intensity values at b^{th} block, we can estimate the background probability as follows:

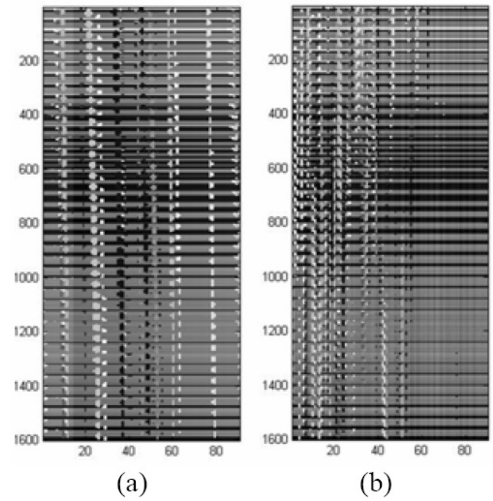

Fig. 1. 90 frames of an instance 40×40 block of a movie with some moving objects: (a) in the temporal order, and (b) in QR decompostion's R-values order.

$$P(B|Y^b_{i,j}) = \begin{cases} 1 & \text{if } j > (1-\beta) * t; \\ 0 & \text{otherwise.} \end{cases} \quad j = 1, \ldots, t \tag{5}$$

where β shows the percentage of the blocks containing background data only. Since the block sizes are small, each block shows nothing but background data in many image frames. Based on our experiments, 1/3 is a proper value for β.

Figure 1 demonstrates the sorting result based on R-values on 90 frames of a 40×40 test block. Fig. 1(a) shows matrix A and Fig. 1(b) shows the same matrix where its columns are sorted based on QR decomposition's R-values (Y series). As can be seen, those columns containing only the background are shifted to the end.

Our previous work [8] had some limitations:

1) The algorithm was unable to update the background model in an online video image. It was an offline algorithm and applying it every few frames to calculate new background model causes an overhead in the processing time.
2) It used a single Gaussian kernel. Therefore, it cannot model the background properly if distribution of the background intensity values does not fit into a single model.

In the rest of this section, we propose a modified version of our QR decomposition based background modeling algorithm in [8] in order to overcome the above problems.

B. Online QR Decomposition Background Modeling

In the proposed algorithm, we split the input frame into M small blocks first. We then partition these M blocks into G groups and in each frame, we only update the blocks belong to one group. Consequently, after G frames, the whole background model will be updated. The following example clarifies this approach: suppose the image size is 120×180 and block size is 30×30, hence we will have $4\times6 = 24$ blocks. We group

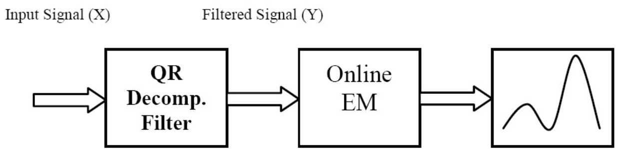

Fig. 2. QR decomposition acts as a filter, which passes only the background pixels to online EM

these blocks into 8 groups: {1,9,17}, {2,10,18}, {3,11,19}, {4,12,20}, {5,13,21}, {6,14,22}, {7,15,23}, {8,16,24} and update one group (3 blocks) in each frame. Therefore we will have less processing tasks for each frame and hence we can maintain the real time performance. We call G as Update Frame Interval (UFI). Increasing UFI will speed up the algorithm; however, it causes the sensitively to background changes to be decreased.

C. QR-MoG Background Modeling

We combine the online approach mentioned in the previous section with the Stauffer's MoG method [1]. We use online version of EM [11] for updating Guassian kernel's parameters. In contrast to the original MoG method, instead of updating the kernels for all pixels, we only consider those pixels detected by the QR decomposition as background pixels. Here, the QR decomposition acts as a filter that allows only background pixels to be used in kernel updating. Although small number of foreground pixels might pass this filter, they will not have much impact on the kernels.

Figure 2 demonstrates the idea of combining these two methods. If we assume the values of a particular pixel over time, $X = \{X_1, X_2, \cdots, X_n\}$ as input signal to QR decomposition, and $Y = \{Y_1, Y_2, \cdots, Y_m\}$ as the output, since QR decomposition only passes the background data, Y will be a subset of X and therefore $m \leq n$. As a result, the online EM algorithm will be executed fewer times and consequently, the resulting hybrid algorithm will run faster than conventional MoG.

In the rest of this section we will show that under two assumptions, the resulting hybrid algorithm will run faster than conventional MoG.

Suppose that $\tau^n_{MoG}$is the processing time of online EM for approximating the Mixture of Gaussian parameters of an input signal $X = \{X_1, X_2, \cdots, X_n\}$, $\tau^n_{QR}$is the processing time of our QR decomposition background modeling algorithm, and $Y = \{Y_1, Y_2, \cdots, Y_m\}$ is the output of the QR decomposition algorithm.

Assumption 1. τ^n_{QR} is a linear function of n: $\tau^n_{QR} = \gamma.n$, where γ is the required processing time for computing QR decomposition of a specified buffer.

According to the mentioned algorithm in sub-section A and using a fixed buffer size, the above assumption is trivial. Since the buffer size is fixed in a single run, and QR decomposition algorithm depends only on the buffer size, the overall processing time, τ^n_{QR}, will be proportional to n, with a constant factor γ [1].

Assumption 2. $\tau^n_{MoG} \propto \tau^n_{QR}$ for a specified buffer size.

Both algorithms are deterministic and average times taken by running each of them on a specified machine, are constant.

Based on our experimental results, for a 40×40 block and buffer length equal to 60, $\tau^n_{MoG} \approx 3\tau^n_{QR}$. If we assume that $\alpha > 0$ is the proportion coefficient, we can write[2]:

$$\tau^n_{MoG} \approx \alpha\tau^n_{QR} \tag{6}$$

Lemma1. With regarding Assumption1 and Equation 6, if $m \leq \frac{\alpha-1}{\alpha}n \Rightarrow \alpha m \leq (\alpha-1)n$ then $\tau^n_{QR} + \tau^m_{MoG} \leq \tau^n_{MoG}$; when $\alpha > 0$ is the proportion coefficient of processing times of the two algorithms, and n, m are the input and output length of QR decomposition portion of the hybrid method, respectively.

Proof:

According to the Lemma's assumption we have:
$m \leq \frac{\alpha-1}{\alpha}n \Rightarrow \alpha m \leq (\alpha-1)n$
After multiplying the above inequality by γ:
$\alpha\gamma m \leq (\alpha-1)\gamma n$.
With regarding the assumption 1: $\tau^n_{QR} = \gamma.n$, we have:
$\alpha\tau^m_{QR} \leq (\alpha-1)\tau^n_{QR} \Rightarrow \tau^n_{QR} + \alpha\tau^m_{QR} \leq \alpha\tau^n_{QR}$
Replacing $\alpha\tau^m_{QR}$ with τ^m_{MoG} and $\alpha\tau^n_{QR}$ with τ^n_{MoG} , based on (6), yields:
$\tau^n_{QR} + \tau^m_{MoG} \leq \tau^n_{MoG}$ and the proof is complete. $\diamondsuit$

Corollary 1: As we see in the above lemma,m, the number of background pixels which are passed by QR decomposition to EM algorithm, must be less than or equal to $\frac{\alpha-1}{\alpha}n$. Because n is the input signal length, m/n is the percentage of the values of a specified pixel that represents background in the image sequence. We named this proportion as β in the previous subsection. Thus $\beta = m/n$ must be less than or equal to $\frac{\alpha-1}{\alpha}$.

Corollary 2: For a specified $\beta = \beta_0$, if $\alpha \geq 1/(1-\beta_0)$ then the lemma 1 assumption: $m \leq \frac{\alpha-1}{\alpha}n$ is satisfied.

Proof:

$\alpha \geq 1/(1-\beta_0) \Rightarrow \alpha(1-\beta_0) \geq 1 \Rightarrow \alpha\beta_0 \leq \alpha - 1 \Rightarrow \frac{m}{n} = \beta_0 \leq \frac{\alpha-1}{\alpha} \Rightarrow m \leq \frac{\alpha-1}{\alpha}n$ and the proof is complete.$\diamondsuit$

From Corollary 2 it is obvious that if $\beta = 1/3$, then α must be greater than or equal to 1.5. It means that $\tau^n_{MoG} \approx \alpha.\tau^n_{QR} \geq \frac{3}{2}\tau^n_{QR}$, or in other words: $\tau^n_{QR} \leq \frac{2}{3}\tau^n_{MoG}$.

As mentioned earlier, our experimental results with $\beta = 1/3$ showed $\alpha \approx 3$; and hence it is sufficient for the proposed hybrid method to be faster than the MoG method.

III. Experimental Results

We applied our proposed background modeling method on test video images and compared its performance with other

[1]Actually γ is the time taken of computing QR decomposition for an specified buffer. Hence γ and τ^n_{QR} are related to *buffSize* and should be written as $\gamma_{buffSize}$ and $\tau^n_{QR,buffSize}$ but for simplicity we drop their subscripts. Experimental results on an Intel Duo Solo Processor T2400 1.83 GHz, showed that γ=0.0063 second for a 160×60 buffer.

[2]Again α is related to *buffSize* and should be written as $\alpha_{buffSize}$, but for simplicity we drop its subscript.

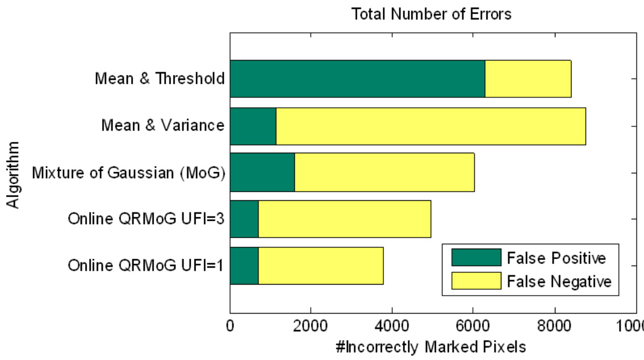

Fig. 3. Overall performance (number of false positive and false negative) of our proposed method in compare with others.

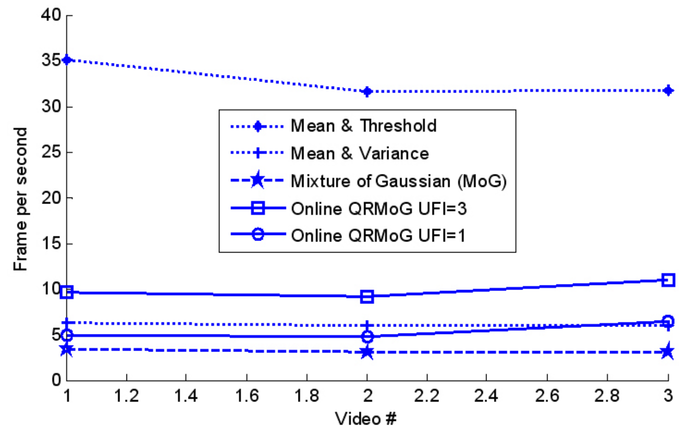

Fig. 4. Speed comparison of different algorithms.

algorithms. We used identical parameters for all methods and applied the same code of standalone MoG in our hybrid method. We computed the mean and variance of the latest 60 frames in the video data for Mean & Threshold, and Mean & Variance algorithms, same as what we did in our proposed method.

The test results are shown in figures 3, 4 and 5. All the algorithms were simulated using MATLAB. Figure 3 illustrates the quantitative comparison between the proposed method and others on false negatives (the number of foreground pixels that were missed) and false positives (the number of background pixels that were marked as foreground). The results of this figure are based on the first three video data shown in Figure 4 which their "Ground truth" foreground pixels were available.

The speed comparisons of tested algorithms are listed in Figure 4. As can be seen from figures 3 and 4, although increasing the UFI will improve the speed, false results will be increased as well. Nevertheless, all the variation of our proposed algorithm has better performance than MoG method.

Figure 5 demonstrates snap shots of the background subtraction results of our proposed method compared with others on the 4 test video data. To have better comparison, no post processing filter was applied to the output of the algorithms. The first video data is a synthetic movie of moving balls over a pure static background image. The image size of the first and the forth video data is 120×160, and we used the block size of 40 for these two videos. The image size of the second and the third videos is 120×180 and we used the block size of 30 for these two videos. From figures 3 and 5, it can be seen that the proposed QRMoG method with UFI=1 has the best performance among the others. However, it is almost two times slower than QRMoG with UFI=3, as depicted in Figure 4.

IV. Conclusion

In this paper, a new method for background modeling based on QR decomposition technique was proposed. We can summarize the advantages of the proposed method in compare with other background subtraction techniques as follows:

- The algorithm can model the background by analyzing few video frames even if there are foreground objects in each frame.
- Mixture model for background image pixels.
- Since the background pixels are recognized before using MoG, the problem of distinguishing which kernel of MoG belongs to background and which is that of the foreground is eliminated.

We also provided a mathematical proof in this paper that how proposed hybrid QR-MoG algorithm runs faster than conventional MoG algorithm.

The experimental results on foreground detection showed better performance and higher processing speed of the proposed method with respect to some other methods. The proposed method can be also used as an initialization step in a hybrid method with other algorithms.

References

[1] C. Stauffer and E. Grimson. "Adaptive background mixture models for real-time tracking", In CVPR, pp 246-252, 1999.

[2] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. "Pfinder: real-time tracking of the human body", In IEEE Trasactions on Pattern Analysis and Machine Intelligence, vol 19, no 7, pp.780-785, 1997.

[3] D.S.Lee, "Effective Gaussian Mixture Learning for Video Background Subtraction", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 27, no 5, pp. 827-832, 2005.

[4] A. Elgammal, D. Harwood, and L. Davis. "Non parametric model for background subtraction", in Computer Vision ECCV 2000, pp. 751-767, 2000.

[5] F. Porikli, O. Tuzel, "Bayesian background modeling for foreground detection", In Proceedings of the third ACM international workshop on Video surveillance & sensor networks, pp. 55-58, 2005.

[6] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, A. K. Jain, "A background model initialization algorithm for video surveillance", In IEEE Int. Conf. on Computer Vision, 2001.

[7] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," In Proc. 13th Conf. Uncertainty in Artificial Intelligence, 1997.

[8] M. Amintoosi, F. Farbiz, M. Fathy, M. Analoui1, N. Mozayani," QR Decomposition-based blgorithm for background subtraction", IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), pp. 1093-1096, 2007.

[9] M. Setnes, R. Babuska, "Rule base reduction: some comments on the use of orthogonal transforms", IEEE Trans. Syst., Man, Cybern, Part C, vol. 31, no. 2, pp. 199-206, May 2001.

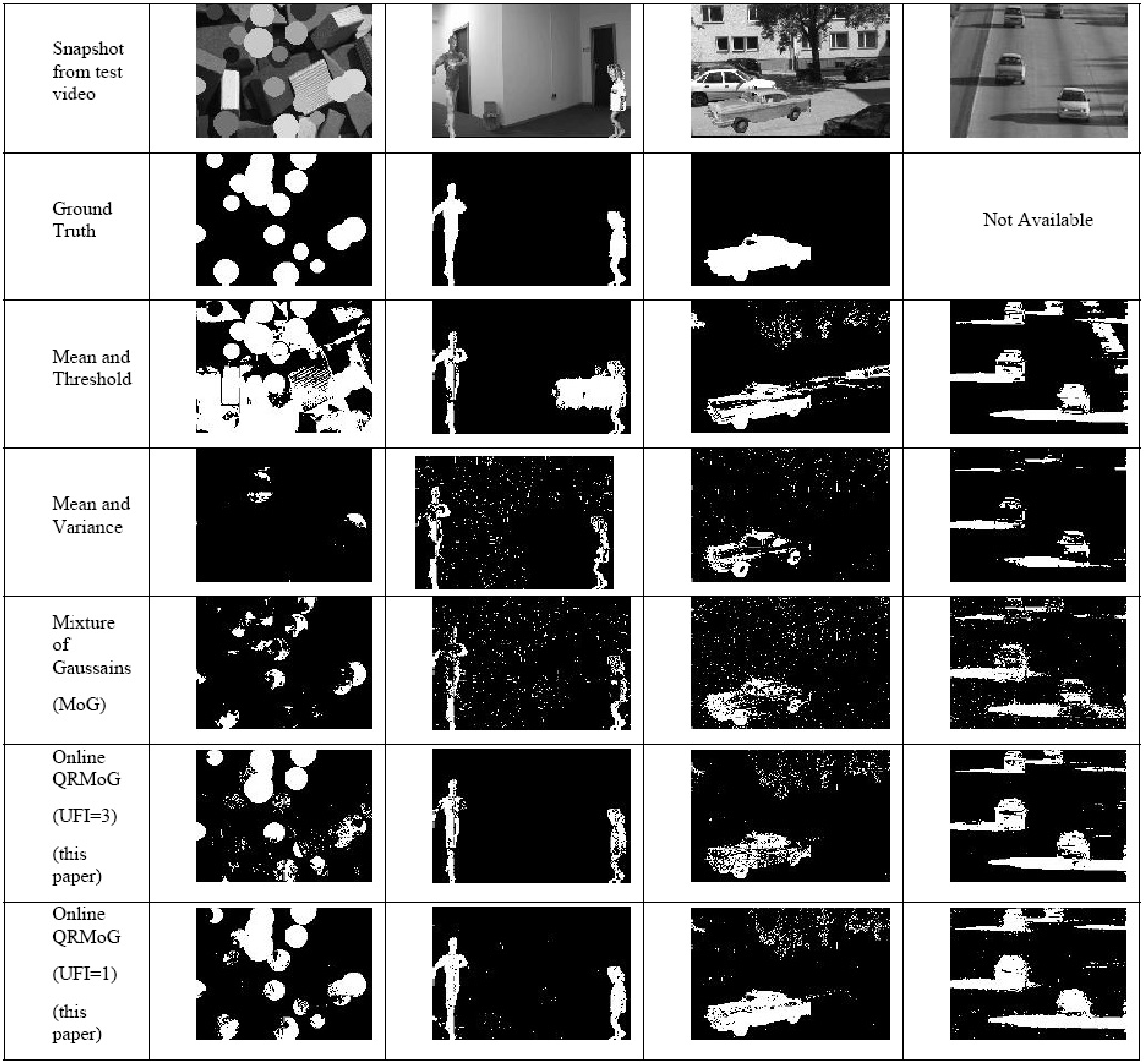

Fig. 5. Result of different background modeling algorithms for foreground detection on four selected test videos.

[10] O. Kaynak, et al., ”Complexity reduction of rule based models: a survey”, In Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, pp. 1216-1221, 2002.

[11] C. Priebe, ”Adaptive mixtures,” Journal of the American Statistical Association, vol. 89, no. 427, pp. 796-806, 1994.