Comparative Evaluation of Background Subtraction Algorithms for High Performance Embedded Systems

Lorena Guachi, Giuseppe Cocorullo, Pasquale Corsonello, Fabio Frustaci, Stefania Perri
Department of Informatics, Modeling, Electronics and System Engineering
DIMES - University of Calabria
Arcavacata di Rende, Italy
e-mail: loreanggeles@hotmail.com, g.cocorullo@unical.it, p.corsonello@unical.it, ffrustaci@deis.unical.it, perri@dimes.unical.it

Abstract— **Background Subtraction technique is widely used in surveillance systems to identify moving objects. Although color features have been extensively used in several background subtraction algorithms, demonstrating high efficiency and performances, in actual real-time applications the background subtraction performance is still a challenge due to high computational requirements. In this paper, two approaches and their optimized versions are evaluated to implement high-performance background subtraction algorithms for real-time applications. Gaussian Mixture Model and the Multimodal Background Subtraction are characterized by two different color descriptors: Gray scale and H color invariant combined with Gray scale information respectively. Different experimental analysis allows evaluating the efficiency in terms of computational complexity and accuracy for outdoor and indoor environments. Experimental tests demonstrated that the Multimodal Background Subtraction approach with its variants is established as affordable for real-time applications and particularly suitable on hardware platforms with on-board memory and limited computational resources.**

Keywords- Real-Time; Image processing; Background subtraction; Segmentation.

I. INTRODUCTION

In recent decades, great interest has been shown for Background Subtraction (BS) technique to achieve a precise pixel classification as background (static) and foreground (dynamic) and then to identify the objects of interest [1] within observed scenes. Since cameras are less expensive than most other sensors and they are already installed on security environments, video sequences are used to build intelligent surveillance systems [2], where many BS algorithms work for specific environments in very controlled situations. Unfortunately, several applications are too slow to be practical as a consequence of their high computational requirements.

The BS algorithms typically use five features as descriptor: color, edge, motion and texture features [3]. Each one is particularly robust to handle critical issues in a different way. For instance, color feature is highly discriminative but depends on the way of representing colors in the image. Therefore, different color representations obtain different accuracies, which are limited in the presence of shadows, illumination changes, and camouflage [1]. On the other hand, edge feature is very discriminative in the presence of ghost and illumination variations. Texture feature works well with shadows and illumination variations, while stereo is robust in order to handle the camouflage issue. Finally, motion feature is useful for detecting articulated objects, but at the expense of increased the computational cost [4].

In order to be more robust in the presence of critical situations, some algorithms combine different features. Therefore, the best solution should reach higher accuracy to classify correctly a pixel as background or foreground. Moreover, it should achieve high speed to incorporate changes from the environment with the ability to run in real-time (RT) without demanding high computational capabilities. In this context, the multi-scale region BS algorithm [5] performs the Gaussian Mixture modeling (GMM) in conjunction with color histograms, texture information, and consecutive division of image regions to efficiently detect edges of the moving objects. Also, in [6], the use of color and edge information is applied to handle slow illumination changes and camera noise, being able to run on standard platform for RT applications.

Although numerous BS algorithms have been introduced with demonstrated efficiency, RT applications, mainly for surveillance systems, remain challenging. One of the reasons is that more robust algorithms usually perform complex operations, thus requiring higher computational capabilities; as a consequence, they are not suitable for RT applications, where portability, low weight, low size, low computational load and low power consumption are required. On the contrary, lower computational loads are usually related to simple background models that lack adaptive background updates and sensitivity to even small background changes.

This paper presents a comparative evaluation of two light and efficient BS algorithms for RT applications oriented to hardware friendly implementations. GMM [7] uses Gray scale and takes advantage of exploiting a color space that does not require complex color transformations. Meanwhile, the Multimodal Background Subtraction (MBSCIG) algorithm [8] exploits two simple background models separately build for the color invariant H and the Gray scale pixels intensities. Experimental tests demonstrate that MBSCIG with its optimized variations can reach higher percentages of correct classified pixels with a reduced computational complexity.

The rest of this paper is organized as follows. Section II describes the most relevant related works. Section III introduces the color descriptors. We briefly explain the

GMM algorithm and its optimized version in Section IV. MBSCIG and its variations are presented in Section V. Section VI presents comparison results, and conclusions are finally drawn in Section VII.

II. RELATED WORKS

In the last years, many different BS algorithms have been introduced, and nearly each of them can provide improvements over the basic algorithms and among each other. They can range from very simple algorithms, usually providing poor performance, to more robust algorithms, which commonly are unsuitable for RT applications due to their high computational complexity. For instance, the Running Gaussian Average [9] uses three color channels for background modeling and models each pixel of each color channel with single Gaussian distribution. The GMM method is exploited in several state-of-the-art algorithms, such as [10-15], to achieve more robustness against frequent and small illumination changes. These algorithms model the history of each pixel over the time by the mean and variance values of a fix number of Gaussian distributions.

The Kernel Density Estimation (KDE) [16] was originally presented by Elgammal like a non-parametric approach to cope with the drawbacks of manually tuning. After that some enhancements have been proposed to decrease the computational complexity using techniques such as histogram approximation and recursive density estimation [17]. The algorithm presented in [18] quantizes each background pixel into codebooks, which represent a compressed form of background model for a long image sequence and are composed of one or more codewords. This allows capturing structural background variation due to periodic motion over a long period of time under limited memory and can handle scenes with moving background, shadows and highlights.

The K-mean algorithms proposed in [19-21] model each pixel of the generic input frame by a group of clusters that are sorted in order of the likelihood to deal with lighting variations and dynamic background. Incoming pixels are analyzed against the corresponding cluster group and are classified according to whether or not the analysis cluster is considered as a part of the background. A fuzzy inference for thresholding is proposed in [22] and [23] in order to improve the thresholding technique avoiding the empirical selection of threshold values by trial and error approach.

In [24], a neural network architecture is proposed to model background images for object segmentation based on an unsupervised Bayesian classifier. The approach proposed in [25] is based on self-organizing through artificial neural networks. It can handle the bootstrapping problem, dynamic scenes containing moving backgrounds, gradual illumination variations and camouflage, which can be included into the background model shadows that cast by moving objects, thus achieving robust detection for different types of videos taken with stationary cameras.

In order to present the aids and constraints of methods based on spatial correlation, density estimates, parametric and non-parametric models, comprehensive reviews are reported in [14], [26] and [27], where the algorithms are evaluated in terms of precision, speed and memory requirements (critical features for RT applications). Concentrated in mathematical models and the solution for critical situations, the author in [3] provides a classification of the traditional and recent works.

To improve stability, accuracy and efficiency, and to support RT applications, a dynamic multi-level feature grouping [2] can be exploited. It introduces the BS and corner cue to detect and handle various sizes of moving objects. To cope the presence of shadows and shading, a basic statistical background modeling at pixel-level is presented in [28] and [29]. However, a dynamic background cannot be handled efficiently with a single-model, especially at the beginning, where the slow learning does not allow differentiating the moving objects from the moving shadows. To solve these limitations, adaptive BS methods are proposed in [30-32]. The latter can efficiently handle quick illumination changes, moving backgrounds and shadow removal.

Additionally, several original methods have been established. As an online estimation of the background in a linear regression, the model demonstrated in [33] achieves high efficiency, while categorizes the foreground as outliers and considers that the background pixels are based on low rank subspace. Parallel analysis at pixel level, presented in [34], holds for each pixel historical and occurrence background values, thus being suitable for both software and hardware implementations. The spatial probability is used in [35], where the eigen background builds the background reference image from a training set of background frames. Based on local texture patterns, the SILTP descriptor is enhanced in [36] to segment the image sequences across of the spatial and temporal analysis of neighborhood. PBAS algorithm [37] relies on the local decision thresholds to segment the foreground, modeling the background with an array of historical frames and choose randomly the observed background pixel to be replaced with the current value.

The most popular algorithms model the temporal video sequences as a parametric form across the Mixture of Gaussians. Such probabilistic technique is shown in [11], where a learning training is required ahead to detect the motion and the interaction between multiple moving objects in the presence of slow light variations and suddenly background changes. A classification of the methods that use the Mixture of Gaussians for foreground detection has been presented in [38], discussing challenges, issues to reduce the computational load, improvements and critical situations that they claim to handle. Based on the remarkable GMM results, in [7] a hardware implementation was proposed for the OpenCV version of the GMM algorithm, and tunings to minimize the word length of the signals able to run on RT applications was performed.

Reached performance by BS algorithms existing in literature also depends on the exploited colors representation [3], [9], [10], [12], [14], [39], [41]. In fact, the color model can significantly influence the achieved quality. In [42] and [43], it is shown that the usage of YCbCr and HSV color spaces can improve the pixels classification. Whereas [44] demonstrates that using the normalized RGB color components leads to higher overall quality and speed performance than those reachable with the c1c2c3 color representation. In [45], color invariant (CI) expressions have been derived that allow the effects of a large set of disturbing factors, such as illumination, viewing direction, surface orientation and highlights, to be significantly reduced in Computer Vision applications. A way to efficiently exploit CIs in BS algorithms has been investigated in [9], where the background model is built referring to N previous frames; each frame is described by the color invariants H, Wx and Wy, and each pixel is modeled with a single Gaussian distribution. An alternative approach was presented in [10], where mixtures of Gaussians are calculated on both the Gray scale pixels intensity and the color invariant Hx. The two channels are then combined to reduce the number of pixel misclassifications in the presence of shadows, noises and illumination changes. In this context, the useful experimental study introduced in [1] provided a point-of-view to choose the best color combination considering accuracy and channel numbers which can be applied for BS. The results demonstrate that the combination of the CI H with Gray scale achieves higher performance for foreground segmentation for both indoor and outdoor video sequences. Then, to make hardware implementation friendlier, the Author exploited in [8] an approximated formulation for CI H transformation from RGB.

Apart of the color representation adopted by BS algorithms, RT oriented algorithms demand a relatively low computational load and must be highly efficient to detect moving objects in diverse environments at common video sequences rates. Therefore, with the aim of establishing the efficiency of the GMM modifications [7] and the MBSCIG [8] algorithm, which are focused on high-performance for RT segmentation, several experimental analysis have been performed using purpose-written C++ routines, which exploit the OpenCV libraries.

In order to reduce efficiently the computational cost of the MBSCIG algorithm, two alternative updating processes are proposed and described in the following. It is notably that, while original techniques provide high robustness, herein, experimental tests show that good performances can be achieved also with the proposed pixel-by-pixel computational scheme through quite tunings. Additionally, performances reached in terms of accuracy, percentage of correct classification, and computational load are comparable with the GMM algorithm presented in [7].

III. Color Descriptor

Most of the work presented in the literature have demonstrated how the color features interfere with the achieved accuracy, typical descriptors are based on specific spectral information (RGB, HSV, HIS, Gray scale, among others). On the other hand, the CIs are derived from a physical model and can take into account color spectral information and color spatial structure. Therefore, in order to build a robust descriptor, handling the issues of pixel-level analysis, an experimental study was presented in [1], which evaluated the color spaces with properties independent of illumination intensity, reflectance property, viewing direction, and object surface orientation, which are defined as the color invariants [46], in conjunction with Gray scale color model.

TABLE I. Set of Color Invariants

CI	*Definition*
H	$E\lambda / E\lambda\lambda$
N	$(E\lambda\chi \times E - E\lambda \times E\chi) / (E \times E)$
C	$E\lambda / E$
W	$E\chi / E$

A. *Color invariant (CI)*

Any method for describing CI model relies on assumptions about the physical variables involved on photometric configuration [44]. Photometric CIs are characterized as functions of surface reflectance, illumination spectrum and the sensing device, which consider the spatial configuration of color, and also the color spectral energy distribution coding color information [9].

Color invariant properties [46] characterize the image color configuration discounting highlights, shadows, noise and shading. As an example, the Gaussian color model with spectral and spatial parameters is exploited in [9] to define a framework for the robust measurement of colored object reflectance.

The CIs are derived from a physical reflectance model based on the Kubelka-Munk theory for colorant layers [45], where illumination and geometrical invariant properties depend on the use of reflectance model. The invariants are useful for materials as dyed paper and textiles, paint films, opaque plastics, dental silicate cements and up to enamel. The CIs derived from Kubelka-Munk theory are listed in Table I. The latter shows how computing the CIs named H, N, C, and W, with E, $E\lambda$ and $E\lambda\lambda$ being the spectral differential quotients based on the scale-space theory [47]. The CIs defined in Table I can be combined incrementally to achieve an alternative to invariant features extraction [44].

B. *Gray scale*

The Gray color space model is based on the brightness information and uses the measurement of amount of light (intensity). It is applied for object tracking often on a blob or a specific region [48]. However, taking into account that the color furnishes more information on the objects in a scene, it would be expected that this model can be used in conjunction with other models to achieve more robust solutions and higher accuracy in comparison with the basic separated models. For this reason, the Gray color space computed by (1) is included in the proposed evaluation to take the advantage of using a color space that does not require complex color transformations.

$$GS=0.299R + 0.58G + 0.114B \qquad (1)$$

IV. GAUSSIAN MIXTURE MODEL

The statistical Background Modeling presented in [12] uses the Gaussian Mixture Model (GMM) to handle efficiently dynamic background. The reported GMM algorithm heads the effectiveness in RT applications, with a good deal between constraints of low computational load and memory requirement, robustness and the ability to cope critical situations, like illumination variation and introduced or removed objects. The improvements of this approach included in the OpenCV library are shown in the following. Some optimizations have been introduced in [7] to obtain efficient hardware implementations. They are cited in the text as "GMM optimized".

A. GMM implemented in OpenCV

The GMM algorithm operates on the probability of observing one process more than one time over a video sequence [10], [12], and assumes that the set of background pixels is visible more frequently than any set of foreground pixels. Based on [12], the algorithm implemented in OpenCV considers that each pixel of each input frame in the video sequence is modeled using *K* mixture of Gaussian distributions in terms of the mean (μ), weight (w), variance (σ^2), and matchsum (counter introduced in OpenCV). Additionally, Fitness (*F*) is used as a sorting parameter to arrange in decreasing order the *K* distributions, and α_w is the learning rate.

To update the background model, each new pixel value (x_t) is checked with respect to *K* Gaussian distributions, calculating the difference between them. If at least one mean difference is less or equal than 2.5σ ($|x_t - \mu_{k,t}| \leq 2.5\sigma$), then the distribution is updated as given in the following equations:

$$\alpha_{k,t} = \alpha_w / w_{k,t} \tag{2a}$$

$$\mu_{k,t+1} = \mu_{k,t} + \alpha_{k,t}(x_t - \mu_{k,t}) \tag{2b}$$

$$\sigma_{k,t+1}^2 = \sigma_{k,t}^2 + \alpha_{k,t}[(x_t - \mu_{k,t})^2 - \sigma_{k,t}^2] \tag{2c}$$

$$w_{k,t+1} = w_{k,t} - \alpha_w . w_{k,t} + \alpha_w \tag{2d}$$

$$matchsum_{k,t+1} = matchsum_{k,t} + 1 \tag{2e}$$

Otherwise, the distribution with the lowest Fitness value is replaced with a new one, for which the mean is set to the current pixel value, whereas the variance and the weight are set to predetermined high variance (*highV*) and low weight (*lowW*), respectively, as shown in the equations below.

$$\mu_{k,t+1} = x_t \tag{3a}$$

$$\sigma_{k,t+1}^2 = high\ v \tag{3b}$$

$$w_{k,t+1} = low\ w \tag{3c}$$

$$matchsum_{k,t+1} = 1 \tag{3d}$$

After the updating step, the weights are normalized so that their summation becomes 1. For each acquired frame at time t, the *K* distributions are sorted in decreasing order of *F* defined in (4).

$$F_{k,t} = w_{k,t} / \sigma_{k,t} \tag{4}$$

To establish whether x_t is part of the background, the first *n* sorted distributions that satisfy equation (5) are selected as background components, and a pixel that matches one of these components is classified as background pixel. In the opposite case, x_t is classified as foreground. The Threshold (*T*) is a fixed value, ranging between 0 and 1, which determines the portion of the distribution weights that defines the background model. Preliminary tests demonstrated that, for the video sequences selected as the benchmarks, *T*=0.75 is the best value.

$$B = arg_n \min \left(\textstyle\sum_{k=1}^{n} w_{k,t} > T\right) \tag{5}$$

B. GMM Optimized (GMM v1)

The GMM algorithm implemented in Open CV is able to work with one or three channels, and its execution involves floating point operations, thus becoming a complex statistical model that provides good accuracy at the expense of a high computational cost, which compromises its use in RT applications. Therefore, in order to reduce the computational cost, the authors proposed in [7] some optimizations based on the following characteristics:

- Handle the algorithm processing with video frames in Gray scale.
- Use fixed-point values for mean (µ) and variance (σ) instead of floating-point values, thus reducing the computational complexity. In fact, floating-point operations use more internal circuitry and require at least 32-bit data paths to manage two parts: the 24-bit integer value (base of the real number) and the 8-bit exponent.
- Establish the word length for each parameter, to reduce the error rate due to the diminution of number of bits.
- Set the number of mixture of Gaussian distributions to *K*=3 as suggested in [34].
- Quantize the learning rates α_w and $\alpha_{k,t}$ as power of two.

$$\alpha_w = 2^{n_w} \quad \alpha_{k,t} = 2^{n_{k,t}} \tag{6}$$

- Use the parameter $IF_{k,t}$, defined in (7) as the square of the inverse of $F_{k,t}$, to sort the Gaussian distributions.

$$IF_{k,t} = (1/F_{k,t})^2 \tag{7}$$

In terms of learning rates, $IF_{k,t}$ is defined as follows:

$$IF_{k,t} = \sigma_{k,t}^2 . 2^{2(n_{k,t} - n_w)} \tag{8}$$

where $n_{k,t} = log_2(\alpha_{k,t})$ and $n_w = log_2(\alpha_w)$.

V. MULTIMODAL BACKGROUND SUBTRACTION MBSCIG

A multimodal BS algorithm has been recently proposed for high performance embedded system MBSCIG [8], with the aim of achieving low computational complexity and high efficiency for RT applications, exploiting the advantage of use a reduced number of channels and historical frames. Only two separate color channels are used to model the Background: one of them is characterized by Gray scale information (G), and another one corresponds to Color Invariant (CI) H. A short detail of the algorithm MBSCIG and its modifications to improve performances are described in the following.

A. *MBSCIG*

MBSCIG gives an effective and quite method using only a modeled frame *mF*, and a small set *hF* of history observations. This approach firstly processes the captured RGB frame to get the Gray scale and H information as describes [8], then it processes the first N+1 acquired frames. The algorithm starts to measure for each pixel of each *hF* the percentage variation *DD* with respect to the current frame I_t. When *DD* is lower that a given Threshold *T*, the counter λ is increased by one. Whether λ counts at least two and the percentage variation *DD* between *mF* and I_t is lower than *T*, the pixel is classified as a background pixel. Otherwise, it is recognized as belonging to the foreground. This analysis is executed for both the channels H and G, computing λh and λg, respectively. As the next step, *mF* is updated as given in equations (9) and (10), depending on the current pixel has been classified as background or foreground. Finally, the oldest frame in *hF* is replaced by I_t .

$$BG_{t+1} = (1-\propto)\ I_t + \propto\ BG_{t+1} \tag{9}$$

$$FG_{t+1} = \beta\ I_t + (1-\beta)\ FG_{t+1} \tag{10}$$

B. *MBSCIG Optimized*

We analyze two alternative ways to perform the updating step of the algorithm MBSCIG. In the original algorithm the background and the foreground are updated as shown in Figure 1a. With the target of limiting the number of operations and reducing the computational load, in order to incorporate gradual changes quickly in the background model, the first alternative approach, reported in Figure 1b, updates the foreground pixels with the value of the current pixel, when the percentage variation is higher than *T*. The second proposed approach, shown in Figure 1c, does not perform any updating operation when a pixel belongs to the set of moving objects.

VI. EXPERIMENTAL RESULTS

Since the learning rate (α) has a fundamental impact on the overall classification in algorithms based on GMM, establishing an appropriate value of α is crucial to achieve high performance with the lowest overall error. Therefore, values in the range [0.01 ÷ 0.05] are evaluated in [49]. In order to select the ideal learning rate value for all tested video sequences, providing good classification, in this work performances achieved are measured not only for α in the range [0.01 ÷ 0.05], but for α equal to 0.1 and 0.005, as suggested in [49] and [50]. The F1 metric is computed for five benchmark video sequences. The F1, introduced in [51] and defined in (11), combines Recall and Precision metrics, defined in (12) and (13), to measure an overall quality of the BS based on True and False Positive and Negative (TP, TN, FP and FN) classifications. The results summarized in Figure 2 show that, when $\alpha = 0.05$, F1 differs from the average of only ±3.3.

capture the current frame
For each pixel *It*(x,y) in the frame
…
if (*DD*<*T* and λ>=*2*)
IsFg=0 //a background pixel is detected
$BG_{t+1} = (1-\propto) .\ I_t + \propto .\ BG_{t+1}$
else
IsFg=1 //a foreground pixel is detected
$FG_{t+1} = \beta . I_t + (1-\beta) .\ FG_{t+1}$
…
End for

a)

capture the current frame
For each pixel *It*(x,y) in the frame
…
if (*DD*<*T* and λ>=*2*)
IsFg=0 //a background pixel is detected
$BG_{t+1} = (1-\propto) .\ I_t + \propto .\ BG_{t+1}$
else
IsFg=1 //a foreground pixel is detected
if (DD > T)
$FG_{t+1} = I_t$
…
End for

b)

capture the current frame
For each pixel *It*(x,y) in the frame
…
if (*DD*<*T* and λ>=*2*)
IsFg=0 //a background pixel is detected
$BG_{t+1} = (1-\propto) .\ I_t + \propto .\ BG_{t+1}$
else
IsFg=1 //a foreground pixel is detected
…
End for

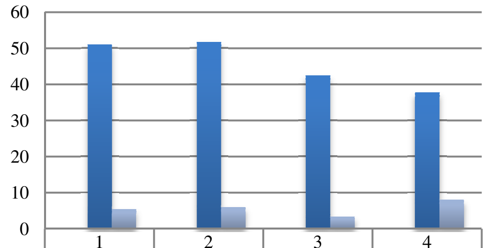
c)

Figure 1. The updating process of the MBSCIG: a) original version; b) MBSCIG v1; c) MBSCIG v2

$$F1 = (2 \times P \times R)/(P + R) \tag{11}$$

$$Recall\ (R) = TP/(TP + FN) \tag{12}$$

$$Precision\ (P) = TP/(TP + FP) \tag{13}$$



	1	2	3	4
Alpha	0.005	0.01	0.05	0.1
F1	51.04	51.69	42.38	37.72
var	5.33	5.98	3.33	7.99

Figure 2. Learning rate performance in GMM

This suggests that using $\alpha = 0.05$, as proposed in [13], is well suited for all tested sequences and can be applied in both indoor and outdoor environments to achieve good object identification.

The versions of GMM and MBSCIG presented in this paper were tested on I2R [52], Wallflower [53], 2012 and 2014 dataset [54]. Lobby is part of I2R dataset, which is defined by illumination changes and complex background, and contains twenty ground-truth images for evaluation target. Wallflowers Dataset includes video sequences with dynamic motions and movement of background objects, such as Waving Trees, which we used in tests considering its ground-truth provided. 2012 and 2014 Datasets contain outdoor and indoor environments, respectively, where Bootstrapping is evaluated based on its one ground-truth, while Office and Highway video sequence have been tested comparing the segmented results with respect to ten ground-truth given.

TABLE II. AVERAGE OF FALSE POSITIVE AND FALSE NEGATIVE RATE

Algorithm	Looby		Waving Tree		Bootstrap		Highway		Office	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
GMM [12]	**0,64**	**1,02**	**0,28**	18,14	**2,08**	14,33	**0,32**	5,47	**0,26**	7,06
GMM v1 [7]	0,71	1,07	10,98	25,17	4,80	14,37	1,45	5,87	2,80	4,71
MBSIG [8]	0,87	1,23	33,18	9,69	7,15	8,46	1,48	4,39	1,16	6,90
MBSIG v1	1,07	1,19	32,88	**7,85**	6,54	6,70	2,16	**3,16**	2,42	3,16
MBSIG v2	7,73	1,21	23,62	8,02	18,97	**4,88**	2,46	3,37	2,73	**1,50**

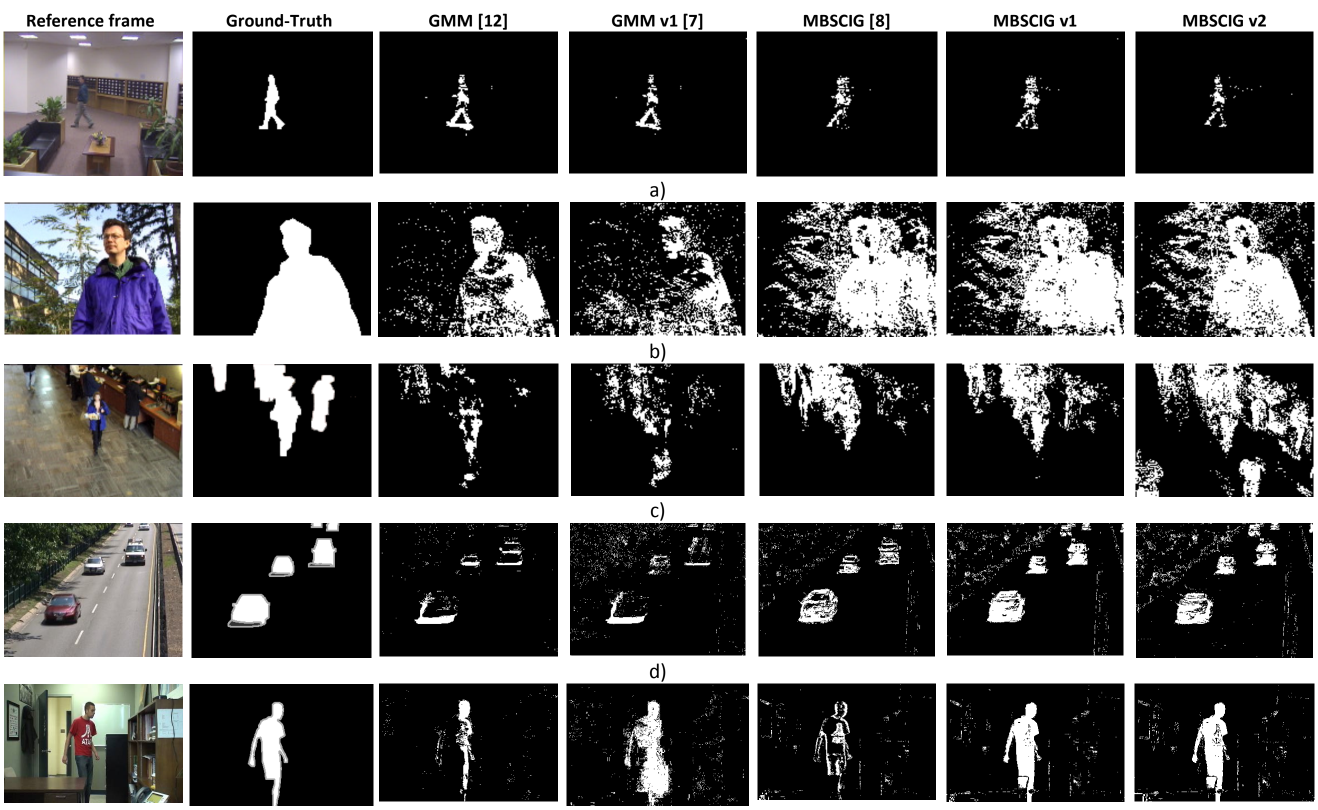


Figure 3. Results for the a) Looby; b) Waving Trees; c) Bootstrapping; d) Highway; and e) Office video sequences.

TABLE III. QUANTITATIVE ACCURACIES.

Algorithm	Lobby		Waving Tree		Bootstrap		Highway		Office	
	F1	**PCC**	**F1**	**PCC**	**F1**	**PCC**	**F1**	**PCC**	**F1**	**PCC**
GMM [12]	**47,62**	**98,37**	67,40	**82,54**	30,71	86,08	38,96	94,67	31,27	93,32
GMM v1 [7]	43,57	98,25	51,22	74,92	27,30	83,74	28,91	93,27	49,21	93,10
MBSCIG [8]	33,58	97,93	61,66	70,27	54,93	86,77	48,46	94,60	30,49	92,63
MBSCIG v1	34,17	97,78	64,06	71,74	**62,99**	**88,78**	**58,74**	**95,09**	**77,68**	**96,41**
MBSCIG v2	20,18	91,21	**69,55**	78,05	52,31	79,78	55,66	94,63	75,03	96,00

TABLE IV. COMPUTATIONAL LOAD

	Color Model	**# Channels**	**Size**	**Background Model**	**Foreground Segmentation**	**Total**
GMM [12]	Gray Scale	1	K=3	(27AS+21MD) x Np	2AS x Np	(29AS + 21MD) x Np
GMM v1 [7]	Gray Scale	1	K=3	(30AS+33MD) x Np	2AS x Np	(32AS + 33MD) x Np
MBSCIG [8]	Gray Scale+H (CI)	2	N=4	(8AS+8MD) x Np	(18AS + 20MD) x Np	(26AS + 28MD) x Np
MBSCIG v1	Gray Scale+H (CI)	2	N=4	(4AS+4MD) x Np	(18AS + 20MD) x Np	(22AS + 24MD) x Np
MBSCIG v2	Gray Scale+H (CI)	2	N=4	(4AS+4MD) x Np	(18AS + 20MD) x Np	(22AS + 24MD) x Np

C++ software routines using OpenCV library have been implemented to evaluate the algorithms. In order to evaluate the performance reachable, for each analyzed algorithm the average of the numerical results achieved processing the selected video sequences has been computed for the evaluated metrics. Table II presents the percentage of False Positive (FPR: Percentage of misclassified pixels detected as foreground) and False negative Rate (FNR: Percentage of misclassified pixels detected as background) defined in (14) and (15). It can be seen that the GMM algorithm obtains the lowest FPR for all the examined video sequences, since it processes only the Gray scale features, which leads to less classification errors. It can also be observed that the FNR takes advantage of the appropriate tuning of the updating process in the MBSCIG algorithm. This is the effect of the modified updating process applied to the foreground pixels, in order handle the sensitivity to small and fast background changes. In fact, the FNR is significantly reduced in Waving Tree, Bootstrap and Highway sequences.

$$FPR = FP/(FP + TN) \tag{14}$$

$$FNR = FN/(TN + FP) \tag{15}$$

Figure 3 illustrates qualitative results for reviewed and optimized BS algorithms. From Figure 3b, we can see that the original version of GMM works better than other algorithms in dynamics backgrounds with small movements. However, the use of only three Gaussian Mixtures in both versions, diminishes the overall accuracy in all experiments. On the other hand, the variants of the MBSCIG algorithm perform much better than original MBSCIG, but all of them are still weak against the dynamic backgrounds.

To present the quantitative accuracy of the tested methods, our experiments compare F1 and Percentage of correct classification (PCC) using equations (11) and (16).

$$PCC = TP + TN/(TP + TN + FP + FN) \tag{16}$$

The results reported in Table III confirm that the variants of the MBSCIG algorithm are robustly capable of detecting moving objects. While, the original GMM algorithm [12] implemented in OpenCV is robust when operating in environments with illumination changes and quick small movements introduced in the background.

Figure 4 plots the F1 average and the percentage of variation of PCC with respect to original version of GMM, and demonstrates that the change in updating process of MBSCIG gives the highest overall accuracy (F1=59.53) with the lowest variation in PCC (only 1.04%).

The computational load of the evaluated algorithms is presented in Table IV separately for the segmentation and the modeling steps in terms of Additions-Subtractions (AS) and Multiplications-Divisions (MD). Also, the number of pixels Np within each Frame is taken into account with the number of channels, and the number of distributions (K) or of historical frames (N). Figure 5a shows that the higher computational load of GMM does not ensure the higher accuracy scores in terms of F1 and PCC metrics. On the contrary, Figure 5b shows that the tuning of the MBSCIG algorithm maintains low values of both FPR and FNR reducing the computational load. From the accuracy and the computational complexity analysis, we can observe that the conjunction between H and Gray scale provides a soft and efficient method with a low computational load.

The variants here proposed for the MBSCIG algorithm have been hardware implemented referring to the system architecture proposed in [8]. The 85K Logic Cells xc7z020 FPGA chip, used to process RGB QQVGA (128×160 pixels per frame) video sequences, allows a 154Mhz running frequency to be reached. Resources requirements are summarized in Table V. It can be seen that the proposed variants occupies less LUTs due to the simplified updating process. Table V also shows that, at a parity of the frame resolution, the hardware designs exhibit computational times reached more than 132 times lower than the pure software executions when performed by one of the Cortex A9 cores running at 800 MHz clock frequency available within the chosen device.

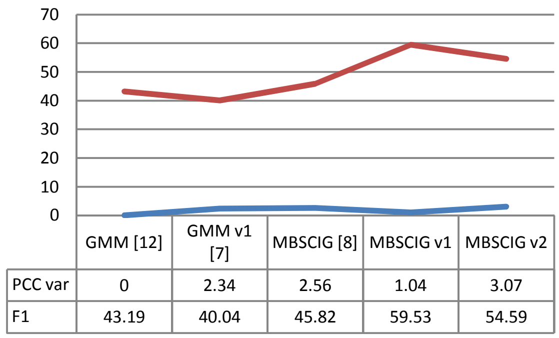


Figure 4. Average and percentage variations of F1 and PCC.

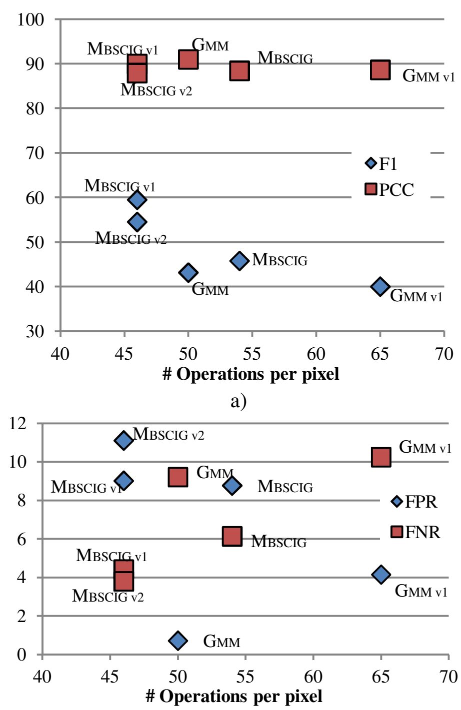


Figure 5. Accuracy vs complexity

TABLE V. HARDWARE DESIGNS VS PURE SOFTWARE EXECUTIONS

	Hardware designs		Software Design
	Resources	Time	Time
MBSCIG [8]	75 BRAM 1868 LUTs 1376 FFs	~0.13ms	~17ms
MBSCIG v1	75 BRAM 1523 LUTs 1376 FFs	~0.107ms	~14ms
MBSCIG v2	75 BRAM 1408 LUTs 1376 FFs	~0.107ms	~14ms

VII. CONCLUSIONS

We have tested two efficient real-time approaches for BS. Based on accuracy metrics we can see that the efficiency in terms of FPR, FNR and F1 are very closer between GMM implemented in OpenCV and MBSCIG with their variations. However, considering the high robustness as the convergence between a good effectiveness with a low computational cost, we can see that MBSCIG and their variations are affordable for real-time applications, and particularly suitable on hardware platforms with on-board memory and limited computational resources and FPGA-based hardware accelerators. As another advantage, the parameters used by the MBSCIG algorithms can be properly chosen, during the design phase, based on preliminary tests performed on video sequences that are typical of the actual scene where the embedded system should work. The adaptability of the algorithms, as well as their performance scalability with video frames of different resolution, will be investigated in future works.

REFERENCES

[1] L. Guachi, G. Cocorullo, P. Corsonello, F. Frustaci, and S. Perri, “Color Invariant Study for Background Subtraction,” in *CENICS 2016 : The Ninth International Conference on Advances in Circuits, Electronics and Micro-electronics*, pp.1-5, 2016.

[2] Z. Kim, “Real Time Object Tracking based on Dynamic Feature Grouping with Background Subtraction,” 2008.

[3] T. Bouwmans, “Traditional and Recent Approaches in Background Modeling for Foreground Detection : An Overview,” *Comput. Sci. Rev.*, pp. 31–66, 2014.

[4] D. Park, C. L. Zitnick, D. Ramanan, and P. Dollar, “Exploring weak stabilization for motion feature extraction,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, no. c, pp. 2882–2889, 2013.

[5] P. Darvish, Z. Varcheie, M. Sills-lavoie, and G. Bilodeau, “A Multiscale Region-Based Motion Detection and Background Subtraction Algorithm,” *Sensors, 2010*, pp. 1041–1061, 2010.

[6] S. Jabri, Z. Duric, and H. Wechsler, “Detection and Location of People in Video Images Using Adaptive Fusion of Color and Edge Information.”

[7] M. Genovese and E. Napoli, “ASIC and FPGA implementation of the gaussian mixture model algorithm for real-time segmentation of high definition video,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 3, pp. 537–547, 2014.

[8] P. Corsonello, G. Cocorullo, F. Frustaci, L. Guachi, and S. Perri, “Multimodal Background Subtraction for high performance embedded systems,” *J. Real-Time Image Process.*, 2016.

[9] H. Zhou, Y. Chen, and R. Feng, “A novel background subtraction method based on color invariants,” *Comput. Vis. Image Underst.*, vol. 117, no. 11, pp. 1589–1597, 2013.

[10] L. Guachi, G. Cocorullo, P. Corsonello, F. Frustaci, and S. Perri, “A novel background subtraction method based on color invariants,” in *Security Technology (ICCST), 2014 International Carnahan Conference on. IEEE*, 2014, pp. 1–5.

[11] C. Stauffer and W.E.L. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, n°8 vol. 22, pp. 747–757, 2000.

[12] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” *Proc. 1999 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Cat No PR00149*, vol. 2, no. c, pp. 246–252, 1999.

[13] J. Sep and S. A. Velast, “F1 Score Assesment of Gaussian Mixture Background Subtraction Algorithms Using the MuHAVi Dataset,”

Imaging Crime Prev. Detect. (ICDP-15), 6th Int. Conf. on. IET, pp. 1–6, 2015.

[14] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Underst.*, vol. 122, pp. 4–21, 2014.

[15] V. J. Butler, Darren and Sridharan, Sridha and Bove, "Real-time adaptive background segmentation," *Acoust. Speech, Signal Process. 2003. Proceedings.(ICASSP'03). 2003 IEEE Int. Conf.*, vol. 3, pp. III–349–52, 2003.

[16] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1162, 2002.

[17] J. Lee and M. Park, "An Adaptive Background Subtraction Method Based on Kernel Density Estimation," *Sensors*, vol. 12, pp. 12279–12300, 2012.

[18] K. Kim, T. H. T. H. Chalidabhongse, D. Hanuood, L. Davis, and D. Harwood, "Background modeling and subtraction by codebook construction," *Int. Conf. Image Process. ICIP '04.*, vol. 5, pp. 3061–3064, 2004.

[19] D. E. Butler and V. M. B. Jr, "Real-time adaptive foreground / background segmentation," *EURASIP J. Appl. Signal Processing*, vol. 2005, no. 14, pp. 2292–2304, 2005.

[20] T.-H. Nguyen, Van-Toi and Vu, Hai and Tran, "An Efficient Combination of RGB and Depth for Background Subtraction," *Springer*, vol. 341, no. January, pp. 49–63, 2015.

[21] P. Parmar, G. Sunilkumar, and P. Jain, "Performance Analysis and Augmentation of K-means Clustering , based approach for Human Detection in Videos," vol. 3, no. 2, pp. 1029–1035, 2015.

[22] X. Lijun, "Moving object segmentation based on background subtraction and fuzzy inference," *2011 Int. Conf. Mechatron. Sci. Electr. Eng. Comput.*, pp. 434–437, 2011.

[23] S. S. Dhar Joydip, Kurele Ritika, Arora Surbhi, "Background Subtraction in Surveillance systems- A Neural Fuzzy Approach," *Int. J. Imaging Robot.*, no. April, 2015.

[24] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "Neural network approach to background modeling for video object segmentation," *Neural Networks, IEEE Trans.*, vol. 18, no. 6, pp. 1614–1627, 2007.

[25] L. Maddalena and A. Petrosino, "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications," vol. 17, no. 7, pp. 1168–1177, 2008.

[26] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art," *Recent Patents Comput. Sci.*, vol. 1, no. 1, pp. 32–54, 2010.

[27] M. Piccardi, "Background subtraction techniques : a review *," pp. 3099–3104, 2004.

[28] M. Sankari and C. Meena, "Estimation of Dynamic Background and Object Detection in Noisy Visual Surveillance," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 6, pp. 77–83, 2011.

[29] E. Hayman and J. Eklundh, "Statistical Background Subtraction for a Mobile Observer," no. Iccv, 2003.

[30] P. Kaewtrakulpong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real- time Tracking with Shadow Detection 2 Background Modelling," pp. 1–5, 2001.

[31] Y. Tian, M. Lu, and A. Hampapur, "Robust and Efficient Foreground Analysis for Real-time Video Surveillance," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference*, 2005, pp. 1182–1187.

[32] S. Huwer, "Adaptive Change Detection for Real-Time Surveillance Applications," no. July, pp. 37–45, 2000.

[33] E. Signal and P. Conference, "An Online Background Subtraction Algorithm Using A Contiguously Weighted Linear Regression Model", University of Kent , UK Pontificia Universidad Católica del Perú , Peru," vol. 1, no. d, pp. 1890–1894, 2015.

[34] B. Wang and P. Dudek, "A Fast Self - tuning Background Subtraction Algorithm," pp. 4321–4324.

[35] L. Vosters, S. Caifeng, and G. Tommaso, "Real-time robust background subtraction under rapidly changing illumination conditions." Image and Vision Computing 30.12 (2012): 1004-1015.

[36] H. Wu, N. Liu, X. Luo, and J. Su, "Real-time background subtraction-based video surveillance of people by integrating local texture patterns," pp. 665–676, 2014.

[37] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background Segmentation with Feedback : The Pixel-Based Adaptive Segmenter."

[38] T. Bouwmans, F. El Baf, B. Vachon, T. Bouwmans, F. El Baf, B. Vachon, B. Modeling, T. Bouwmans, F. El Baf, and B. Vachon, "Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey To cite this version : Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey," 2008.

[39] J. M. Guo, Y. F. Liu, C. H. Hsia, M. H. Shih, and C. S. Hsu, "Hierarchical method for foreground detection using codebook model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 6, pp. 804–815, 2011.

[40] V. Reddy, C. Sanderson, and B. C. Lovell, "Improved foreground detection via block-based classifier cascade with probabilistic decision integration," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 1, pp. 83–93, 2013.

[41] O. Barnich and M. Van Droogenbroeck, "ViBe : A universal background subtraction algorithm for video sequences," *Image Process. IEEE Trans.*, vol. 20, no. June, pp. 1709–1724, 2011.

[42] S. R. Sivanantham S, Nitin Paul, "Object Tracking Algorithm Implementation for Security Applications," *Far East J. Electron. Commun.*, vol. 16, no. 1, p. 17654, 2016.

[43] R. H. Luke, S. Member, D. Anderson, S. Member, and J. M. Keller, "Human Segmentation from Video in Indoor Environments Using Fused Color and Texture Features." Technical Report, University of Missouri. 2007.

[44] B. Shoushtarian and H. E. Bez, "A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking," vol. 26, pp. 5–26, 2005.

[45] J. Geusebroek, R. Van Den Boomgaard, I. C. Society, A. W. M. Smeulders, S. Member, and H. Geerts, "Color Invariance," vol. 23, no. 12, pp. 1338–1350, 2001.

[46] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," *Pattern Recognit.*, vol. 32, no. 3, pp. 453–464, 1999.

[47] M. A. Florack, Luc MJ and ter Haar Romeny, Bart M and Koenderink, Jan J and Viergever, "Scale and the differential structure of images," *Image Vis. Comput.*, vol. 10, pp. 376–388, 1992.

[48] P. Sebastian, Y. V. Voon, and R. Comley, "Colour Space Effect on Tracking in Video Surveillance," vol. 2, no. 4, pp. 298–312, 2010.

[49] S. S. Mohamed, N. Tahir, and R. Adnan, "Background modelling and background subtraction performance for object detection Background Modelling and Background Subtraction Performance for Object Detection," in *Signal Processing and Its Applications Conference*, 2010.

[50] A. Bouzerdoum and S. L. Phung, "On the analysis of background subtraction techniques using Gaussian mixture models," pp. 4042–4045, 2010.

[51] P. Goyette, Nil and Jodoin, Pierre-Marc and Porikli, Fatih and Konrad, Janusz and Ishwar, "Changedetection . net : A New Change Detection Benchmark Dataset," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 1–8, 2012.

[52] "Statistical Modeling of Complex Background for Foreground Object Detection." [Online]. Available: http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html. [Accessed: 23-Sep-2016].

[53] B. M. Kentaro Toyama, John Krumm, Barry Brumitt, "Test Images for Wallflower Paper," 1999. [Online]. Available: http://research.microsoft.com/en-

us/um/people/jckrumm/wallflower/testimages.htm. [Accessed: 23-Sep-2016].

[54] "ChangeDetection.Net (CDNET) A video database for testing change detection algorithms." [Online]. Available: http://www.changedetection.net/. [Accessed: 23-Sep-2016].