Asian Journal of Scientific Research

ISSN 1992-1454

science alert
http://www.scialert.net

ANSInet
an open access publisher
http://ansinet.com

OPEN ACCESS Asian Journal of Scientific Research

ISSN 1992-1454
DOI: 10.3923/ajsr.2017.139.149

Research Article
Implementation of Moving Object Segmentation using Background Modeling with Biased Illumination Field Fuzzy C-Means on Hardware Accelerators

[1]Siva Nagi Reddy Kalli and [2]Bhanu Murthy Bhaskara

[1]Department of Electronics and Communication Engineering, Sridevi Women's Engineering College, Jawaharlal Nehru Technological University, Hyderabad, Telangana, India
[2]College of Computer and Information Sciences, Majmaah University, 15341 Al Majmaah, Kingdom of Saudi Arabia

Abstract

Background and Objectives: The study was focused to make use of committed hardware structural design for moving object segmentation. The competent architecture by the improved performance algorithm to produce accurate results was proposed in this study. The objective of this study demonstrated: (1) Accurate motion object segmentation algorithm intended for video supervision system. (2) Implementation and study of its computational complexity of proposed algorithm architecture on Hardware Accelerators (Field Programmable Gate Arrays and Application Specific Integrated Circuits). **Methodology:** To accumulate the objectives the simulation was conducted to evaluate and generate the accurate measures using the Background Modeling along with Biased Illumination Field Fuzzy C-Means (BM-BIFCM) algorithm. For the examination of the mentioned algorithm performance, the standard video was considered and corresponding values of proposed algorithm was derived using Matlab tool. The architecture implemented on Xilinx Vivado Field programmable gate arrays devices via Very High Speed Integrated Circuit Hardware Description Language or Verilog code in Integrated Software Environment tools fitting and same in Application Specific Integrated Circuits using Cadence tools. **Results:** The effect of the algorithm was demonstrated as considerable proof to boost the correctness of segmentation procedure using metrics and execution on hardware outcome illustrated the complexity of architecture decreased in both Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuits (ASIC). **Conclusion:** The response of the suggested method produced accurate results, so that it may be relevant in real time applications efficiently. The implementation obstacles reduced in the direction of chip area, power and delay on hardware architecture, so that cost of the chip design diminished by using the presented algorithm.

Key words: Motion object segmentation, back ground modeling, biased illumination field fuzzy C-means, stationary pixel, clustering and integrated circuit

Received: November 30, 2016 Accepted: April 14, 2017 Published: June 15, 2017

Citation: Siva Nagi Reddy Kalli and Bhanu Murthy Bhaskara, 2017. Implementation of moving object segmentation using background modeling with biased illumination field fuzzy C-means on hardware accelerators. Asian J. Sci. Res., 10: 139-149.

Corresponding Author: Siva Nagi Reddy Kalli, Department of Electronics and Communication Engineering, Sridevi Women's Engineering College, Jawaharlal Nehru Technological University, Hyderabad, Telangana, India Tel :+919951523234

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

In video surveillance schemes, automatic scene analysis was essential to identify person or object. Now a day these systems can automatically select the frames of concern in a video stream by using scene change motion detection algorithms. The main dispute for designing these real time systems should be positioned in their algorithm and hardware performance constraints. In this study, the projected algorithm has used a hybrid method, which was a pixel-based method while it was simpler and yields more precise results. The hybrid method was the combination of Background Modeling and Biased Illumination Field Fuzzy-C-Means (BM-BIFCM). This method automatically adjusts to different atmosphere along with removes non-background information or adds new-background values and it also rapidly adjust to abrupt or gradual illumination changes. In addition, the scheme uses morphological processing operations, which deals for removing image elements suitable for the demonstration and explanation of shape. General fundamental morphological operations are dilation (expands an image) and erosion (shrinks an image). Most of the existing methods realized in software only and a few methods implemented on hardware in real time. For real-time processing of a video stream for surveillance applications the execution of motion detection algorithm could not achieve better performance on general purpose processors. In order to get higher efficiency, an alternative processor like Field Programmable Gate Arrays (FPGA) and Application Specific Integrated Circuit (ASIC) has been used instead of general purpose processors in this study. Comparing FPGA and ASIC, the FPGAs has capability to permit alteration in the design in later stages of system development, architectural efficiency about high throughput and area. A lot of research effort has been prepared for hardware realization of moving object detection so far. A few design architectures have been discussed in literature in this study.

The hardware realization of image processing system based optical flow method implementation on FPGA[1] with Yosemite sequence of 252×316 size in 47.8 msec .This real time processer was designed for larger images at faster speed of visual motion. An extended optical flow algorithm[2] was introduced, it was an iterative algorithm, its precision depends largely on the number of iterations carried out and it was implemented on FPGA[3]. The traditional Lucas and Kanade approach[4] shaped better accuracy and was also employed[5] for its good processing efficiency. A rapid and exact motion evaluation algorithm established[6], for that an architecture was designed and implemented in FPGA[7]. This hardware setup can process images of size 640×480 at 64 fps. A new background subtraction and foreground tracking hardware-oriented algorithm[8] proposed for targeting SoC architecture with use of single camera. The architecture design consists of two acceleration units and programmable micro processor unit. The projected devise can able to process image size of 352x288 with operating frequency of 30 MHz. To process of high resolution videos with frame size 720×576 at rate of 50 fps, a new hardware realization based design introduced on Pixel-Based Adaptive Segmenter (PBAS) foreground object detection algorithm in FPGA[9]. An embedded automated digital video supervision system[10,11] had presented. This design realized with MoG-based background subtraction method along morphological operations on a Xilinx FPGA platform. The introduced design decreases the memory bandwidth 70% by adopting word-length reduction scheme, but it fails in environments with slow moving objects. An optimized method using adapting GMM background subtraction model implemented on FPGA[12]. This model operates 91 HD fps and it also implemented on ASIC (UMC-90 nm CMOS technology), which created better performance results over the above design models. A simple FPGA implementation of block based mean square error based method[13] for detection of moving object with similar bit width of background and reference frame and implemented in Virtex-6 and Virtex-5 and 4 using Vivado tool and which was also implemented in ASIC provides appreciable results but failed to produce the predictable area constraints. A new framework for real time motion object segmentation was introduced using the Background Model Fuzzy C-Means Algorithm (BMFCM)[14]. It has produced promising results in hardware realization over the supplementary methods.

MATERIALS AND METHODS

The presented video moving object segmentation scheme includes both Background Modeling with Biased Illumination Field Fuzzy-C-Means (BM-BIFCM). This algorithm reduces non stationary pixel from the object and increases the segmentation efficiency. In this proposed system, the moving object detection under the static camera system framework found which lowers the difficulties of the background modeling. Background modeling was the procedure of recognizing the moving objects from the part of a video frame that fluctuates extensively commencing a background scene. The most often challenged drawback in segmenting the foreground object from background scene of a frame and classifying the motion alters of each frame by assess it by means of previous frame had foreground object might modification steady where as background scene maybe static, however, in few videos along the foreground and background

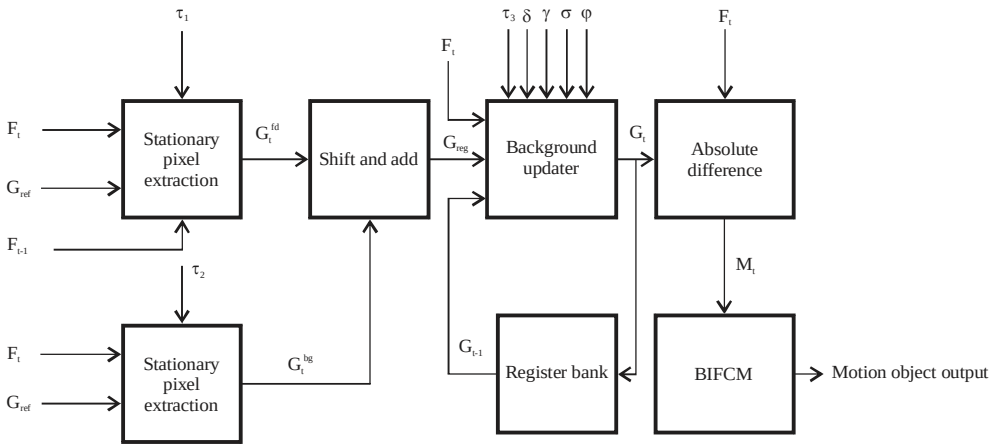


Fig. 1: Hardware design architecture for BM-BIFCM

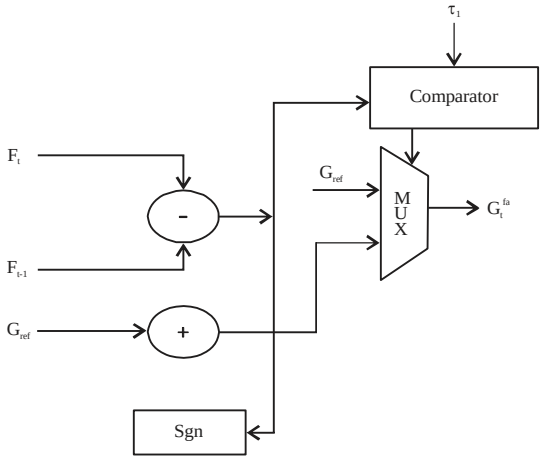


Fig. 2: Hardware structure for stationary pixels through frame difference method

objects might progress[15]. This setting of a video specific to form a noise, hidden edges, loss of smoothness, improper segmentation of foreground object particularly in overlapped objects and maintaining strength because of changes in illumination had troublesome within the implementation of background subtraction algorithms[16].

To resolve such kind of problem in background subtraction, the proposed method uses two stages. The initial stage gives an appropriate background model monitored by an updating scheme. In the second stage, based on identical property the region level dispensation has been done and each and every pixel can be affected independently in a frame using clustering techniques. In the view of hardware realization of the image processing algorithms on hardware accelerators (FPGA and ASIC) occupies more space and time and power consuming. To address all the issue of hardware architecture and observe long term effects from parameter settings in addition to fixed point quantization simulation can be performed with the help of FPGA platform. By using a Xilinx FPGA reconfigurable device the moving object architecture can be implemented. The utilization of FPGA in this design, synthesis and development time can be reduced. The same architecture also implemented and verified the parameters in the form of power, area and delay in ASIC with TMSE 180 nm technology.

Hardware architecture for moving object segmentation: The architecture proposed method includes (1) Stationary pixels or non-stationary pixel extraction $[G_t^{reg}(u, v)]$ via mean of frame difference model $[G_t^{fd}(u, v)]$ and background subtraction method $[G_t^{bg}(u, v)]$ using shift and add circuit, (2) Back ground updated circuit had used to perform the ground truth, (3) Absolute difference circuit provides initial motion field and (4) BIFCM provides final motion object foreground object by eliminating noise via reducing the biased illumination field is shown in Fig. 1.

To perform the update background process "G_t" averaging pixel and some parameters like δ, γ, σ and φ were used in this method. Previous frame "G_{t-1}" also modernize the background through register bank. The absolute difference between the current frame "F_t" and updated background frame "G_t" generates initial motion field.

Background Model generation (BM)

Stationary pixel using frame difference $[G_t^{fd}(u, v)]$: The initial frame and reference background denoted as F_0 (u, v), G_{ref} (u, v), respectively, which contains no foreground object. In this model the static pixels and non static pixel are isolated from the reference background frame and the frame difference.

The hardware architecture of stationary pixel from the frame difference method is shown in Fig. 2, this action can be performed by using of threshold value comparison with frame difference. The set of stationery file was selected by using the

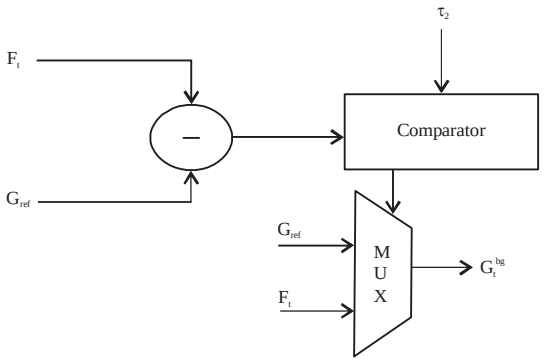


Fig. 3: Stationary pixels hardware structure for background subtraction method stationary pixels

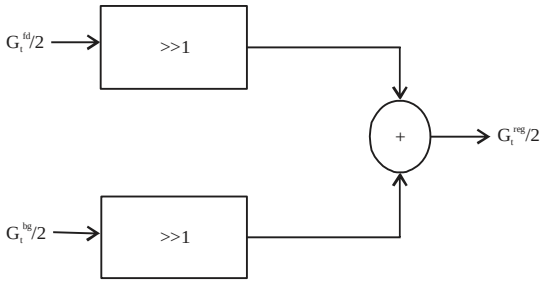


Fig. 4: Single stationary pixels design

difference between the current frame F_t (u, v) and previous frame F_{t-1} (u, v). The reference background frame G_{ref} (u, v) follows:

$$G_t^{fd}(u, v) = \begin{cases} G_{ref}(u, v), \text{ if } |F_t(u, v) - F_{t-1}(u, v)| < \tau_1 \\ G_{ref}(u, v) \times \text{sgn}[F_t(u, v) - F_{t-1}(u, v)], \text{ otherwise} \end{cases} \quad (1)$$

where, G_t^{fd} mentioned as stationary pixels via frame difference model and τ_1 is threshold value. In the hardware circuit MUX helps to select the G_t^{fd} from G_{ref} of background frame regarding the threshold value.

In Eq. 1 the signum function was defined as:

$$\text{sgn}(d) = \begin{cases} 1, \text{ if } d > 0 \\ 0, \text{ if } d = 0 \\ -1, \text{ if } d < 0 \end{cases} \quad (2)$$

where, $d = F_t(u, v) - F_{t-1}(u, v)$ represents the input value.

Stationary pixel using background subtraction [G_t^{bg} (u, v)]: The current input frame F_t (u, v) subtracts from reference background frame G_{ref} (u, v) used to investigate the stationary pixels:

$$G_t^{bg}(u, v) = \begin{cases} G_{ref}(u, v), \text{ if } |F_t(u, v) - G_{ref}(u, v)| > \tau_2 \\ F_t(u, v), \text{ otherwise} \end{cases} \quad (3)$$

where, $G_t^{bg}(u, v)$ represents stationary pixel, which was measured by the background subtraction method and threshold τ_2, respectively.

The hardware design for background subtraction provides the second stationary pixel by using multiplexer (MUX) and comparator shown in Fig. 3.

Mean stationary pixels calculation: Figure 4 provides average of foreground stationary pixel from the frame difference method [G_t^{fd} (u, v)] and background stationary pixel from the background subtraction method [G_t^{bg} (u, v)] given as:

$$G_t^{reg}(u, v) = \frac{G_t^{fd}(u, v) + G_t^{bg}(u, v)}{2} \quad (4)$$

Initial variance modeled as:

$$\sigma_j^2(u, v) = \text{var}(F_0(u, v)) \quad (5)$$

By using initial variance, the current change in spatial variance was given as:

$$\sigma_d^2(u, v) = \sigma_j^2(u, v) + \text{sgn}[\text{var}(F_t(u, v)) - \sigma_j^2(u, v)] \quad (6)$$

where, σ_d^2 (u, v) represented as current spatial variance.

The initial motion field had the difference between background frame and current frame. To the matched pixel and essential magnitude of foreground intensity pixels set initial motion to zero intensity.

Appropriate to similarity of background and foreground pixel, the false negative pixels and holes can be formed in the moving objects. To reduce such kind of problems by exact selection learning rate γ and update the background pixel frame. The current background frame can estimate by Eq. 7:

$$G_t(m, n) = \begin{cases} \gamma.G_{t-1}(u, v) + (1-\gamma).(F_t(u, v) - G_{t-1}(u, v)), \\ \text{if } 0 < |F_t(u, v) - G_{t-1}(u, v)| < 1 \\ \delta.G_{t-1}(u, v) + (1-\delta).(\sigma_d^2(u, v) - \sigma_j^2(u, v)), \\ \text{elseif } |F_t(u, v) - G_{t-1}(u, v)| < \varphi\sigma_d \\ \text{and } 1 < |F_t(u, v) - G_t^{reG}(u, v)| < \tau_3 \\ G_{t-1}(u, v) + \text{sgn}(F_t(u, v) - G_{t-1}(u, v)), \\ \text{else} \end{cases} \quad (7)$$

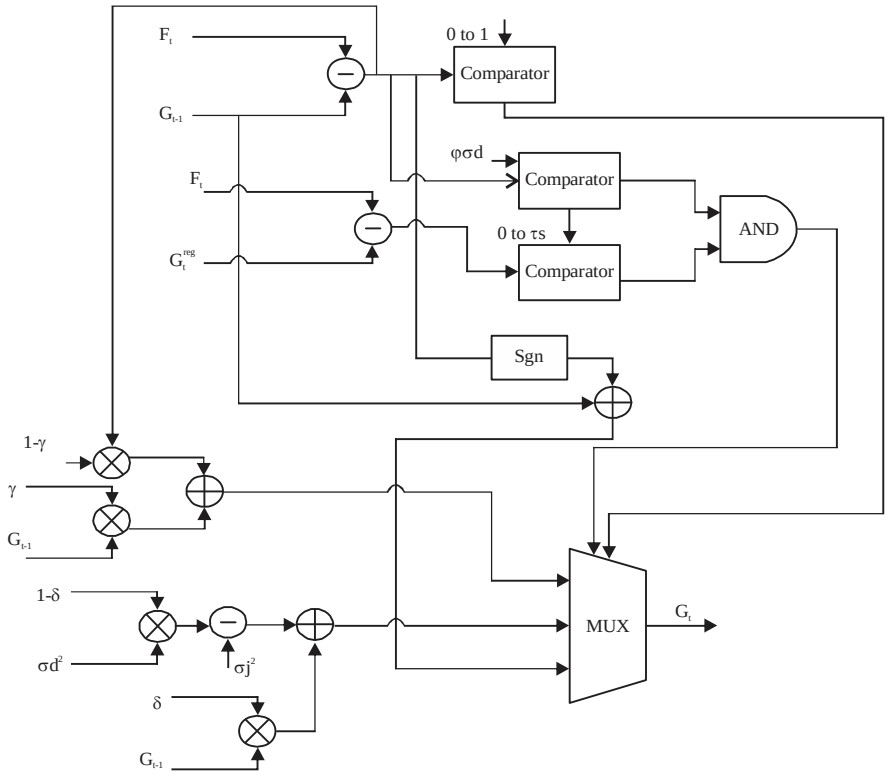


Fig. 5: Updated background estimation circuit architecture

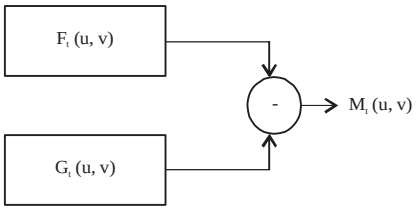


Fig. 6: Motion field estimation circuit design

In Eq. 7, initial reference frame or previous background and current updated background mentioned as $G_{t-1}(u, v)$ and $G_t(u, v)$, respectively. The σ_d and σ_j represent current standard deviation of current frame and initial standard deviation of the reference background frame. The τ_3 represent as threshold value, which was user defined value. The δ, γ and φ values are ranging from 0.8-0.99, 0.999 for all videos and 1-3. The current updated background pixel hardware architecture is shown in Fig. 5.

To update the background pixel model, the current frame integrates with different frames by using a recursive filter, which provides the difference between the background and foreground pixel intensity level. Due to the local motion in the background, the variance of the pixel change causes the specious detection. By using learning rate γ, enhance the initial spatial variance and current pixels to avoid erroneous detection. As a result it produced the false positive pixels.

In this method the initial motion field characterized the absolute difference shown in Fig. 6, it was the difference between the current background and the first frame mentioned as fallows as:

$$M_t(u, v) = F_t(u, v) - G_t(u, v) \quad (8)$$

Clustering algorithm: In higher level applications like moving object detection, the shape of correct object location and noise elimination are key factors. For that principle, the proposed algorithm uses Biased Illumination Field Fuzzy C-Means (BI-FFCM) method and it was an improved version of Modified FCM (MFCM) algorithm[17] and provides finest outcome. The disadvantage of MFCM is that it is not able to produce accurate object function due to errors which come from inhomogeneous illuminations of neighborhood pixels. This can be minimized by BI-FFCM and hence the motion objects segmentation efficiency increase. For finding the clustering object function, the average intensities of pixel neighborhoods could play vital role.

The traditional FCM object function had illustrated by using the following function:

$$J = \sum_{i=1}^{c}\sum_{k=1}^{N} M_{ik}^{w} \left\| u_k - c_i \right\|^2, 1 \le w < \infty \quad (9)$$

where, N represented as the total number of pixels in image, $U = (u_1, u_2 \ldots u_N)$ was the set of pixel intensities, c was the number of classifications, M_{ik} was the membership degree of kth pixel u_k to ith cluster centroid c_i, w symbolized as the exponential weight of membership and $\|u_i - c_i\|$ represented as the Euclidean norm distance between u_k and c_i. When $M_{ik} \in [0, 1]$, $w = 1$ and c_i was the centroid of cluster, $w \in [1, \infty]$ had a weighing exponent:

$$J = \sum_{i=1}^{c}\sum_{k=1}^{N} M_{ik}^{m} \left\| u_k - c_i \right\| \quad (10)$$

Equation 9 and 10 formed nearly the equal results. For simplification purpose linear distance measure was considered in this work from the Eq. 9.

Biased illumination field fuzzy C-means: The fuzzy liner distance measure of Eq. 9 can be rewritten as a modified object function illustrated as:

$$J_m = \sum_{i=1}^{c}\sum_{k=1}^{N} M_{ik}^{w} \left\| u_k - s_k - c_{i1} \right\| + \delta \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{w} \left\| \bar{u}_k - s_k - c_{i2} \right\| + \gamma \left(1 - \sum_{i=1}^{c} M_{ik} \right) \quad (11)$$

In Eq. 11 u_k and $\bar{u}_k$ are the intensity, averaged intensity of the kth pixel's neighborhood. The c_{i1} and c_{i2} are the cluster centers of the unbiased intensities, u_k and the centers of its neighborhood, respectively. Where, s_k represented as optimal estimation of the bias field. The second term in Eq. 11 facilitates the labeling of a pixel to be controlled by its intermediate neighborhood. The control parameter δ was inversely proportional to the signal to-noise ratio of the image. The third term in Eq. 11, γ represented as a Lagrange multiplier and $\sum_{i=1} M_{ik} = 1$. Differentiating J_m with respect to M_{ik} and $\gamma = 0$ for $w = 2$:

$$M_{ik}^{*} = \frac{1}{\sum_{j=1}^{c} \left(\frac{o_{ik} + \delta p_{ik}}{o_{jk} + \delta p_{jk}} \right)^{1/(w-1)}} \quad (12)$$

where, M_{ik}^{*} represented as the optimal membership estimation of the kth pixel belonging to the ith class, $o_{ik} = \|u_k - s_k - c_{i1}\|$ and $p_{ik} = \|\bar{u}_k - s_k - c_{i2}\|$. The derivative of J_m with respect to c_{i1} and c_{i2}, setting $\partial J_m / \partial c_{i1} = 0$, $\partial J_m / \partial c_{i2} = 0$ at $j = 1, 2$, achieved as:

$$\begin{bmatrix} c_{i1}^{*} = \sum_{k=1}^{N} M_{ik}^{w} (u_k - s_k) \Big/ \sum_{k=1}^{N} M_{ik}^{w} \\ c_{i2}^{*} = \sum_{k=1}^{N} M_{ik}^{m} (\bar{u}_k - s_k) \Big/ \sum_{k=1}^{N} M_{ik}^{w} \end{bmatrix} \quad (13)$$

where, $i = 1, 2 \ldots c$.

The optimal estimation of the bias field 's_k' had determined from Eq. 11:

$$s_k^{*} = \frac{\delta}{\delta^2 + \delta} \left\{ (v_k + \delta \bar{v}_k) - \left(\sum_{k=1}^{c} M_{ik}^{w} (c_{i1} + \delta c_{i2}) \Big/ \sum_{k=1}^{c} M_{ik}^{w} \right) \right\} \quad (14)$$

$$\|C_{new} - C_{old}\| < \varepsilon \quad (15)$$

In this study 'ε' value was considered as 10^{-3}. The c_{i1} and c_{i2} are 20,100 with cluster number of $N = 5$.

The background modeling with BIFCM in dynamic scenes for 5 clusters as follows as:

$$D_t(u, v) = \begin{Bmatrix} 1, \text{if } M_{ik}(u, v)_1 > M_{ik}(u, v)_2 \\ 0, \text{if } M_{ik}(u, v)_1 < M_{ik}(u, v)_2 \end{Bmatrix} \quad (16)$$

A morphological filling operation had used to perform the foreground mask and its structuring element considered as 4×4 window with all ones was used. For getting accurate motion segmented video output as follows:

$$H(u, v) = D_t(u, v) \cap \text{hole} \quad (17)$$

RESULTS AND DISCUSSION

For the testing of developed algorithm (BM-BIFCM) mentioned in methodology, a test sequence with outdoor conditions of traffic highway video has been considered in this study. The choice of such different scenes was prepared to highlight the consistency and robustness of the presented method in outdoor circumstances.

A standard performance metrics were considered to analyze the observed pixels about to the ground truth image depend on True Positive (TP) pixels, True Negative (TN) pixels, False Positive (FP) pixels and False Negative (FN) pixels. True Positive (TP) pixels were the correctly detected pixels by the algorithm of the moving object.

The sensitivity standards of the suggested algorithm can find out by the following parameters. The relevant pixels (Recall) and irrelevant pixels (Precision) of the detected object can be initiated as fallows as:

$$\text{Recall} = \frac{TP}{TP + TN} \tag{18}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{19}$$

$$\text{Similarity} = \frac{TP}{TP + FP + FN} \tag{20}$$

$$\text{False measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Precision} + \text{Recall}} \tag{21}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

True positive rate and true negative rate:

$$TPR = \frac{TP}{TP + FN} \tag{23}$$

$$TNR = \frac{TN}{TN + FP} \tag{24}$$

False positive rate and false negative rate:

$$FPR = \frac{FP}{FP + TN} \tag{25}$$

$$FNR = \frac{FN}{TP + FN} \tag{26}$$

The positive predictive value and negative predictive values:

$$PPV = \frac{TP}{TP + FP} \tag{27}$$

$$NPV = \frac{TN}{TN + FN} \tag{28}$$

The algorithm described BM-BIFCM has been experienced with outdoor video stream. The ground truth reference has been prepared for a video stream by physically extracting frame-by-frame of every pixel of each moving vehicle. The simulation results showed that 85-93% of the pixels fit to the moving object in ground truth image are appropriately recognized by the method was Recall (Eq. 18). Additionally, the precision ratio in Eq. 19, specify, how many surrounded detected pixels belong to the moving objects was about 55-57%. Simulation results for numerous frames of the selected video streams are exposed in Fig. 7. The method had a capability to correctly identify the moving object at various scene conditions as shown in Fig. 7. A poorer quality detection results typically arises in: Dark scenes or in strong sun light causing intensive shadows. Comparing all the existing with BM-BIFCM, the developed approach output gives clear object information without noise. For all of the above images are numerically shown in Table 1.

Figure 8 shows robustness of the projected method to hold with outdoor circumstances. Figure 8 consists of background frame, previous frame and current frames are taken as 50th, 119th and 120th frame are reference frames in video. The segmented output and ground truth frame are taken as 150th frame from video delivered the final motion object mask.

The above results showed the potential of the developed advance. It was necessary to estimate the efficiency of the method with a ground-truth. The purpose of the projected method was not only for simply detection improvement or discrimination of shadow pixels; it's for efficient and precise object detection for further relevance. The efficiency of the numerical values of presented algorithm evaluates to the existing methods shows parameters Recall, similarity and F-measure in Table 1, for the given traffic car video. Let us consider the value of precision metric provides 57% with BM-BIFCM method, where as the previous method BMFCM provides the 56%, others fallowed as 51, 47 and 49%. In terms similarity proposed method provides 54.5%, others were 53.8% BMFCM, 47% ICM[19], 44% BMSE and 47% GMM[18]. The metric F-measure of proposed method 70.5% followed other methods as 69.9, 64.65, 61.75% and 64.47. From the Table 1 the developed approach are relatively superior, considering that the contrast connecting the object and background was very low.

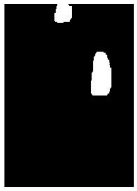
			Motion object segmentation				
Video's	Sample frame	Ground truth	Gaussian mixture model[18]	Block mean square error[13]	Iterated conditional modes[19]	Background model fuzzy C-means[14]	BM-BIFCM (Proposed)
Car traffic							

Fig. 7: Motion mask generated by the proposed method and other baseline methods

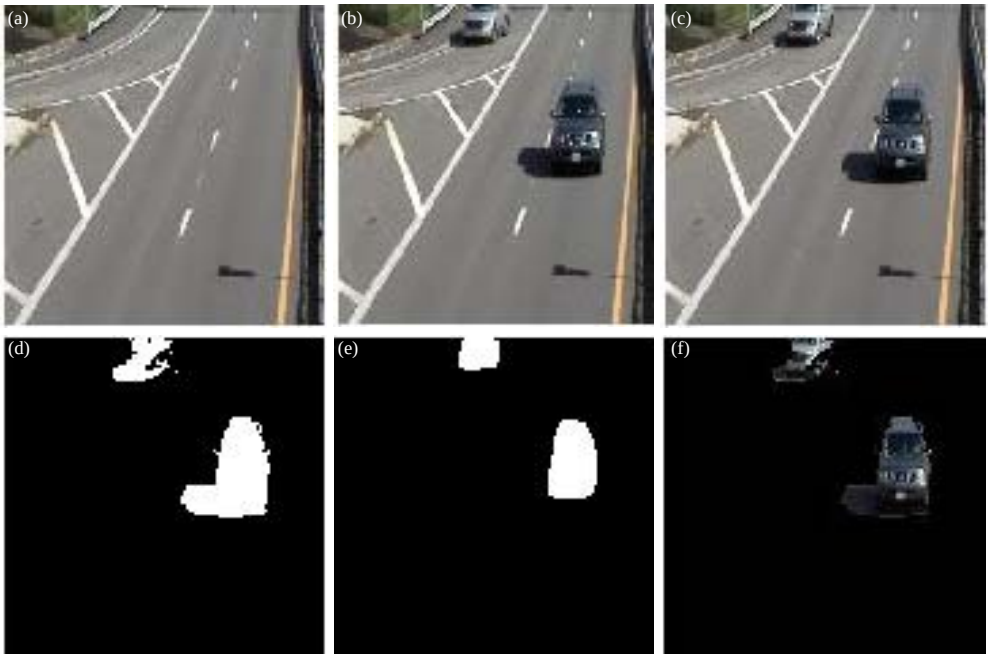


Fig. 8(a-f): Object motion segmentation and mask generation of the background modeling with bias illumination filed FCM method (a) Background frame, (b) Previous frame, (c) Current frame, (d) Segmented frame output, (e) Ground truth frame and (f) Detected object

FPGA implementation of BM-BIFCM

Field-Programmable Gate Array (FPGA): It was a semiconductor device enclosed with programmable logic components called "Configurable logic blocks" and programmable interconnects. The Configurable Logic Blocks (CLBs) are the basic logic unit of an FPGA and it can be programmed to execute the function of basic logic gates for instance AND and XOR or decoders. The modern designed FPGAs, the CLB also contain memory elements as simple Flip-Flops (FFs) or more complete blocks of memory, Look-up tables (LUTs) and multiplexer. Increase in number of CLBs can increase the performance of a FPGA[20].

- **Flip-flop:** It was a circuit accomplished of two stable states that correspond to a single bit and it acted as small storage device. In FPGA circuit every flip-flop in a CLB had a binary shift register utilized to save logic states among clock cycles
- **Look-Up Table (LUT):** It stored a predefined list of outputs for every combination of inputs and which offered a quick approach to recover the output of a logic operation since achievable results are accumulated and after that mentioned quite than estimated
- **Multiplexer (MUX):** A circuit that picks one of several input signals and forwards the selected input into a single line

Table 1: Summary of different method with defending videos

Sequences	Evaluation	Gaussian mixture model[18]	Block mean square error[13]	Iterated conditional modes[19]	Background model fuzzy C-means[14]	BM-BIFCM (Proposed)
Car traffic	Recall	0. 9349	0.9030	0.8763	0.9293	0.9247
	Precision	0.4930	0.4703	0.5132	0.5610	0.5701
	Similarity	0.4767	0.4472	0.4780	0.5380	0.5453
	F-measure	0.6447	0.6175	0.6465	0.6990	0.7049
	Accuracy	0.9714	0.9687	0.9731	0.9776	0.9784
	True positive rate	0.9349	0.9030	0.8763	0.9293	0.9247
	True negative rate	0.9724	0.9707	0.9760	0.9790	0.9798
	False positive rate	0.0276	0.0293	0.1237	0.0210	0.0202
	False negative rate	0.0651	0.0970	0.0240	0.0707	0.0753
	Positive predictive value	0.04930	0.4703	0.5132	0.5610	0.5701
	Negative predictive values	0.9980	0.9969	0.9962	0.9978	0.9978

Table 2: A summary result for proposed architecture and others was implemented on FPGA's

Target FPGA	Circuit	Pipe line levels	Look-up Table	Flip-flop	Slice	DSP-MULT	Block RAM	Frequency (MHz)	Video fps
Virtex4 (xc4vfx12)	BM-BIFCM	0	186/10944	65/10944	89/5472	0/32	16	214.891	30
	BMFCM[14]	0	192/10944	77/10944	108/5472	0/32	16	243.891	30
	BMSE[13]	0	1850/10944	38/10944	1009/5472	9/32	0	111.101	30
	ICM[19]	-	1865/10944	76/10944	960 /5472	27/32	0	10.238	30
Virtex5 (xc5vlx50)	BM-BIFCM	0	153/28800	68/28800	43/7200	0/48	0	259.175	30
	BMFCM[14]	0	162/28800	77/28800	57/7200	0/48	0	272.298	30
	BMSE[13]	0	317/28800	148/28800	132/7200	9/48	2	308.166	30
	ICM[19]	0	1240/28800	72/28800	355/7200	35/48	0	9.279	30
	GMM[11]	1	724/28800	223/28800	323/7200	3/48	0	130.90	63
Virtex6 (xc6vlx75t)	BM-BIFCM	0	1074/46560	65/93120	34/11640	3/288	0	125.492	30
	BMFCM[14]	0	114/46560	71/93120	41/11640	3/288	0	131.376	30
	BMSE[13]	0	261/46560	140/93120	111/11640	9/288	2	282.135	30
	ICM[14]	0	1860/46560	74/93120	540/11640	35/288	0	101.965	30
	GMM[11]	1	788/46560	363/93120	349/11640	3/288	0	189.30	91

GMM : Gaussian mixture model, ICM : Iterated conditional mode, BMSE: Block mean square error, BMFCM : Background modeling fuzzy C-means, BM-BIFCM: Background modeling biased illumination field fuzzy C-means, DSP-MULT: Multiple digital signal processors, FPGA: Field programmable arrays

Table 3: ASIC (TSMC 180 nm technology design) parameters comparisons of proposed architecture with existing state art

Method	Cell area (μm²)	Power (nw)	Delay (ps)
BM-BIFCM	556834.76	1243.00	5737
BMFCM[14]	587091.04	1391.00	6659
Block MSE[13]	1115693	1847.42	14261
ICM[19]	4920659.48	7439.53	102822.40

GMM: Gaussian mixture model, ICM: Iterated conditional mode, BMSE: Block mean square error, BMFCM: Background modeling fuzzy C-means, BMBIFCM: Background modeling biased illumination field fuzzy C-means, ASIC: Application specific integrated circuits

From the above information every CLB had n-inputs and single output either the registered or the unregistered LUT output. The CLB output was selected using MUX. The LUT output had registered using the FF. The clock is given to the flip-flop, using which the output was registered. The clock signals were routed through committed routing set-ups. All the CLBs connected via programmed routing channels, in such a way that logic of the hardware design was realized.

Block RAM (BRAM): It was a type of random access memory specifically fixed all over an FPGA for data storage. The main purpose of BRAM was used to transfer data between multiple clock domains, FPGA targets and store bulky data sets on an FPGA target effectively than RAM created from LUTs. The LUTs, F/F's and BRAM's arranged in slices. It indicated that those elements shares connections in turn to operate fast carry chain.

FPGA programming: Primarily the hardware design was coded in Hardware Descriptive Language (Verilog or VHDL) and code was simulated and synthesized. The synthesis was done using tools like Xilinx ISE and net-list was created, then the net- list targeted to actual FPGA architecture by place and route[20].

The implementation of BM-BIFCM on FPGA as fallows as:

- Analysis of the BM-BIFCM motion object segmentation in Matlab 2015a
- BM-BIFCM algorithm converted into Verilog code
- Targeted Verilog code of segmented algorithm on to Virtex4 (xc4vfx12), Virtex5 (xc5vlx50) and Virtex6 (xc6vlx75t) by using Vivado HLS (2014.2)
- By comparing the Verilog-libraries with HLS libraries, synthesized the Verilog code
- Analyzed the synthesis report and generated the report is shown in Table 2

The presented BM-BIFCM had synthesized and implemented Xilinx (VIVADO) FPGA Virtex 6(xc6vfx20), Virtex5 (xc5vlx50) and Virtex 4(xc4vlx75t), devices. Fitting, Place and Route can be carried out using ISE tool and model-sim had used for simulation circuit. For better performance, the cost of the FPGA should be less and it was measured by using number of flip-flops and slices in the FPGA. Similarly, frequency also important parameters in FPGA for measuring speed. Table 2 illustrates the outcome of the projected hybrid background modeling circuit targeted to FPGA. The proposed architecture realized with reduced amount of chip area with fewer slices used at high frequency.

The design of moving object segmentation modeled in Verilog HDL, synthesized via TSMC 180 nm standard-cells library, placed, routed and chip-finished. This process accomplished by using Cadence Encounter RTL Compiler. The designs have been simulated with NCSim and the Toggle Count File (.tcl) has been generated in order to obtain an exact view of the power dissipation. Table 3 provides cadence synthesis outcomes of design (BM-BIFCM) with the cell area/chip area reduced by almost 10% with BMFCM. The required power for proposed algorithm implementation compared with BMFCM method reduced by 10 % and delay by reduced by 12%. From Table 3, the projected design of ASIC parameters as power consumption and processing delay produces improved results over the existing designs.

CONCLUSION AND FUTURE RECOMMENDATIONS

In this study, the projected method reduced the noise and developed accurate segmentation by reduction of false pixel using hybrid algorithm over the other methods. After realization of the BM-BIFCM algorithm in FPGA achieved real-time capability with 30 fps with frame size of 640×480 in live video and also architecture used less logic resources (>10% total resources). In addition hardware design implemented in ASIC performed significant presentation parameters in terms of area, power and delay. By the observation of the projected method, the efficiency in terms of F-measure, Recall and precision values were not presented much higher due to false negative pixels with neighbor-hood pixel. The computation delay was also more and does not offer automated selection of the clusters centroid, such that it cannot able to provide much hardware efficiency.

SIGNIFICANCE STATEMENTS

- The real time implementation of video motion object segmentation was essential to find identification of persons or objects in shopping mall, airports and railways etc. for safety purpose

- The previous studies fail to give the precise output of the motion object segmentation and also unsuccessful to provide better implementation parameters like memory, chip area and delay on hardware accelerators like field programmable arrays (FPGA) and Application Specific Integrated Circuits (ASIC)
- The presented Background Modeling with Biased Illumination Field Fuzzy C-means (BM-BIFCM) algorithm provides excellent simulation results compared with existing algorithms in addition hardware accelerators (FPGA and ASIC) implementation results provide less resources utilizations

REFERENCES

1. Correia, M.V. and A.C. Campilho, 2002. Real-time implementation of an optical flow algorithm. Proceedings of the 16th International Conference on Pattern Recognition, Volume 4, August 11-15, 2002, IEEE., pp: 247-250.
2. Horn, B.K.P. and B.G. Schunck, 1981. Determining optical flow. Artif. Intell., 17: 185-203.
3. Martin, J.L., A. Zuloaga, C. Cuadrado, J. Lazaro and U. Bidarte, 2005. Hardware implementation of optical flow constraint equation using FPGAs. Comput. Vision Image Understanding, 98: 462-490.
4. Lucas, B.D. and T. Kanade, 1984. An iterative image registration technique with an application to stereo vision. Proceedings of the DARPA Image Understanding Workshop, October 3-4, 1984, New Orleans, pp: 121-130.
5. Diaz, J., E. Ros, F. Pelayo, E.M. Ortigosa and S. Mota, 2006. FPGA-based real-time optical-flow system. IEEE Trans. Circuits Syst. Video Technol., 16: 274-279.
6. Farneback, G., 2000. Fast and accurate motion estimation using orientation tensors and parametric motion models. Proceedings of the 15th International Conference on Pattern Recognition, Volume 1, September 3-7, 2000, IEEE., pp: 135-139.
7. Wei, Z., D.J. Lee, B. Nelson and M. Martineau, 2007. A fast and accurate tensor-based optical flow algorithm implemented in FPGA. Proceedings of the Workshop on Applications of Computer Vision, February 21-22, 2007. IEEE., Austin, Texas, USA., pp: 18-23.
8. Tsai, T.H., C.Y. Lin, D.Z. Peng and G.H. Chen, 2009. Design and integration for background subtraction and foreground tracking algorithm. Proceedings of the 5th International Conference on Information Assurance and Security, Volume 1, August 18-20, 2009, IEEE., pp: 181-184.
9. Kryjak, T., M. Komorkiewicz and M. Gorgon, 2014. Hardware implementation of the PBAS foreground detection method in FPGA. Comput. Sci. Inform. Syst., 11: 1617-1637.
10. Kristensen, F., H. Hedberg, H. Jiang, P. Nilsson and V. Owall, 2008. An embedded real-time surveillance system: Implementation and evaluation. J. Signal Proces. Syst., 52: 75-94.
11. Jiang, H., H. Ardo and V. Owall, 2009. A hardware architecture for real-time video segmentation utilizing memory reduction techniques. IEEE Trans. Circuits Syst. Video Technol., 19: 226-236.
12. Genovese, M. and E. Napoli, 2014. ASIC and FPGA implementation of the gaussian mixture model algorithm for real-time segmentation of high definition video. IEEE Trans. Very Large Scale Integration (VLSI) Syst., 22: 537-547.
13. Kalli, S.N.R. and B.M. Bhaskara, 2016. FPGA implementation of BMSE based motion object segmentation. Int. J. Elect. Elect. Telecommun. Eng., 46: 1560-4564.
14. Kalli, S.N.R. and B.M. Bhaskara, 2017. Efficient field programmable gate array implementation for moving object segmentation using BMFCM. Indian J. Sci. Technol., Vol. 8.
15. Cheung, S.C. and C. Kamath, 2005. Robust background subtraction with foreground validation for urban traffic video. EURASIP J. Applied Signal Process., 14: 1-11.
16. Cheng, L., M. Gong, D. Schuurmans and T. Caelli, 2011. Real-time discriminative background subtraction. IEEE Trans. Image Process., 20: 1401-1414.
17. Ma, L. and R.C. Staunton, 2007. A modified fuzzy C-means image segmentation algorithm for use with uneven illumination patterns. Pattern Recognit., 40: 3005-3011.
18. Kalli, S.N.R. and B.M. Bhaskara, 2016. Image segmentation by using modified spatially constrained gaussian mixture model. Int. J. Scient. Eng. Res., 7: 624-629.
19. Kumar, S. and J. Yadav, 2016. Segmentation of moving objects using background subtraction method in complex environments. Radio Eng., 25: 399-408.
20. National Instruments, 2017. Introduction to FPGA resources. National Instruments. http://www.ni.com/documentation/en/labview-comms/1.0/fpga-targets/intro-fpga-resources/