# REDUCING RUN-TIME FOR MOVING OBJECT DETECTION IN BACKGROUND BUFFERING APPROACH

**A.R. Revathi[1] and Dhananjay Kumar[2]**
[1]*Department of Information Technology, Valliammai Engineering College, India*
E-mail: revathirajendran2010@gmail.com
[2]*Department of Information Technology, Madras Institute of Technology, Anna University Chennai, India*
E-mail: dhananjay@annauniv.edu

***Abstract***
***An efficient moving object Segmentation is useful for real time content based video surveillance and Object Tracking. Commonly a foreground is extracted using a mixture of Gaussian followed by shadow and noise removal to initialise the Object Trackers. This technique uses a kernel mask to make the system more efficient by decreasing the search area and the number of iterations to converge in the new location of the object. In the background model, the post processing step is applied to the obtained object mask to remove noise region and to smoothen the object boundary which incurs additional delay. In this paper a Background Buffering algorithm (BBA) is proposed to construct a reliable background model based on a sequence of input frames. Except moving object data, the other data is used to build reliable background representation. The result shows significant decrease in run-time for the higher level processing steps of surveillance system.***

***Keywords:***
***Object Segmentation, Video Segmentation, BBA Algorithm, Background Registration***

## 1. INTRODUCTION

The active research topics in computer vision are the dynamic scenes detection, classifying object, tracking and recognizing activity and description of behaviour. Visual surveillance strategies have long been in use to gather information about monitoring people, events and activities. The main goal of visual surveillance is not only to monitor, but also to automate the entire surveillance task. The goal of visual surveillance is to develop intelligent surveillance to replace the traditional passive video surveillance that is proving ineffective as the numbers of cameras exceed the capability of human operators to monitor them. The automated surveillance system can be implemented for both offline and online process. Offline process is nothing but analysing the information from the stored video sequence. But nowadays online surveillance system is very much needful in all public and private sectors to predict and avoid unwanted movements like terrorist activities in those areas. The stages of video surveillance include Motion segmentation / object detection [3], object classification, object tracking [2] and behaviour understanding [1]. Background modelling is a widely used approach for detecting moving objects in videos from static cameras. Video surveillance works as follows: detect moving object [4], [5], [6], classify the detected objects [7], [8] and object track [9], [10] them through the sequence of images and analyse the behaviour. Before modelling the background, needs to do pre-processing such as shadow removal in order to eliminate errors or additional information that may propagate through each stage which subsequently affects the final result. Similarly each stage of visual surveillance must produce reliable intermediate result. The result of one stage becomes input for the next stage. So propagation of some unwanted information may affect the performance of the visual surveillance systems. The first and fore-most stage is Motion segmentation/object detection [1], [10]. This involves identification of objects like car, bus or people. The detection of moving [11], [5] object is done by taking the difference between current frame and reference frame often called as "Background Image" or "Background Model".

The background image must be a representation of a scene with no moving objects and must be updated regularly so as to adapt the varying luminance conditions and geometry structures. A model of background scene is built and each pixel in the background image is analyzed. A pixel's deviation in colour or intensity value is used to determine whether the pixel falls either in background or in foreground category. There are various problems [5] associated with background subtraction such as sudden changes in the lighting conditions, moving object present during the initialisation of background scene, shadows present at any time, movement through cluttered areas, [10] objects that overlap in the image (occlusion), effects of moving elements in the background scene (such as swaying tree branches), slow moving objects, if a foreground's pixel characteristics are almost same as the background, etc. several methods to perform background subtraction has been proposed in recent years and all these methods try to efficiently estimate background model from the temporal sequence of the frames. The background subtraction technique used should be carefully chosen according to the scene where the action will take place.

In this paper, we propose an easy to implement and automated background modelling in surveillance system using BBA method. An update mechanism utilising foreground detection is used to initialise buffer with static background pixel. Rest of this paper is organised as follows. The related work for object segmentation is presented in section 2 and section 3, discussion about various approaches in segmentation. In Section 4 and 5, the proposed object segmentation is discussed. Experimental results demonstrating the performance of the proposed algorithm are given in section 6. Finally this paper is concluded in section 7.

## 2. RELATED WORKS

In the literature, a reliable foreground segmentation algorithm that combines temporal image analysis with a reference background image is presented in [1]. They also propose a new approach for background adaptation to the changes in illumination. In [12], a novel non-parametric

background model and a background subtraction approach are presented. This model can handle situations where background of the scene is cluttered and not completely static but contains small motions such as tree branches and bushes. The model in [12] estimates the probability of observing pixel intensity values based on a sample of intensity values of each pixel.

In [13] a survey of original classification of the improvements in Mixture of Gaussians (MoG) is presented. They discuss about the issues related to reduction in computation time and challenges met in video sequences. In [14], the authors evaluate several popular, state-of-the-art Background Subtraction Algorithms (BGS) and examine how post-processing techniques affect their performance. Post processing techniques can significantly improve the foreground segmentation masks produced by BGS algorithms. In [15], authors present an efficient method for recursive density approximation that relies on the propagation of the density modes. At each time step, the modes of the density are re-estimated and a Gaussian component is assigned to each mode. This work adapts a variable-bandwidth mean shift approach and is used to detect the mode.

In [16], the authors present an approach to background modelling based on mean-shift procedure. The mean-shift convergence properties enable the system to achieve reliable background modelling. In addition, histogram based computation and the new concept of local basins of attraction allows meeting the stringent real-time requirements of video processing. In [11], a review of the main methods of background subtraction and an original categorisation based on speed, memory requirements and accuracy is presented.

## 3. VARIOUS METHODS

### 3.1 FRAME DIFFERENCE

The Background can be calculated by subtracting the current frame from the previous frame. This method work well only for static background, particular frame rate and object speed. The disadvantage of this method is that it does not find complete objects, sensitive to small light changes and generates noisy segmentation. It is calculated using the formula

$$B(a,b) = \left| fr_t(a,b) - fr_{t-1}(a,b) > T \right| \quad (1)$$

where, $B(a,b)$ is the calculated background, $fr_t(a,b)$ is the current frame, $fr_{t-1}(a,b)$ is the previous frame and $T$ is the threshold chosen. Another major drawback of this method is choosing value of $T$ threshold and it does not support multimodal background distributions.

### 3.2 RUNNING GAUSSIAN AVERAGE

We have proposed to model the background independently at each ($a$, $b$) pixel location. The model is based on ideally fitting a Gaussian probability density function to the last $n$ pixel's values. In order to avoid fitting the pdf from scratch at each new frame time t, a running (or on-line cumulative) average is computed instead as:

$$\mu_t = \alpha fr_t(a,b) + (1-\alpha)\mu_{t-1} \quad (2)$$

where, $\mu$ the previous average; α is an empirical weight often chosen as a trade-off between stability and quick update. Each background pixel $B_t(a,b)$ is individually and separately modelled. It tries to find average of last n frames. To prevent over fitting of Gaussian distribution (also known as pdf) from scratch for each new frame time $t$, the running Gaussian average is computed as follows:

$$\mu_t(a,b) = \alpha fr_{t-1}(a,b) + (1-\alpha)\mu_{t-1}(a,b) \quad (3)$$

The pixel is classified as either foreground or background based on the following condition:

$$\left| fr_t(a,b) - \mu_{t-1}(a,b) \right| > T \quad (4)$$

This method is well known for its simplicity and less memory requirements. The main disadvantage of this method is that the value of $\alpha$ is chosen randomly (or arbitrarily). It is a single Gaussian PDF and it is insufficient to support multimodal background.

### 3.3 MIXTURE OF GAUSSIAN

Mixture of Gaussians is a widely used approach for background modelling to detect moving objects from static cameras. Once the background model is defined, the different parameters of the mixture of Gaussians must be initialized. The parameters of the MOG's model are the number of Gaussians $K$, the weight $\omega_{i,t}$ associated to the $i^{th}$ Gaussian at time $t$, the mean $\mu_{i,t}$ and the covariance matrix $\sum_{i,t}$. A match is found with one of the K Gaussians [20]. In this case, if the Gaussian distribution is identified as background, the pixel is classified as background otherwise the pixel is classified as foreground. A match is found with one of the $K$ Gaussians. For the matched component, the update is done as follows:

$$\mu_{i,t+1} = (1-\alpha)\omega_{i,t} + \alpha \quad (5)$$

where, $\alpha$ is a constant learning rate.

$$\mu_{i,t+1} = (1-\rho)\mu_{i,t} + \rho X_{t+1} \quad (6)$$

$$\sigma_{i,t+1}{}^2 = (1-\rho\sigma)^2{}_{i,t} + \rho(x_{t+1} - \mu_{i,t+1})^T \quad (7)$$

where, $\rho = \alpha\eta(x_{t+1}, \mu_i, \Sigma_i)$ for the unmatched components, $\mu$ and $\Sigma$ are unchanged, only the weight is replaced by:

$$\omega_{i,t+1} = (1-\alpha)\omega_{j,t} \quad (8)$$

No match is found with any of the K Gaussians. In this case, the pixel is classified as foreground. Match is found with any of the $K$ Gaussians. In this case, the least probable distribution $K$ is replaced by a new one with parameters:

$\omega_{k,t+1}$ = low prior weight

$\mu_{k,t+1} = x_{t+1}$

$\sigma^2{}_{k,t+1}$ = Large Initial Variance

The MOG model deals with the movement in the background (MB) due to the multimodality in the representation step.

### 3.4 KERNEL DENSITY ESTIMATION

An approximation of the background pdf can be given by histogram of the most recent values classified as background values [10]. If we keep track of the intensity values of a single pixel over time and if there is no background movement, the intensity can be modelled with single Gaussian kernel, and the image noise over the same time can be modelled with Gaussian distribution with Zero Mean. The KDE model obtains the most recent information about the image sequence and continuously

updates this information to obtain the fast changes in the background. The intensity distribution of a pixel can change quickly so we estimate this density function of this distribution at any time, by using only the most recent history information. KDE produces smoothed continuous version of histogram. Kernel density estimations (KDE) however runs faster comparatively than MoG, but it still requires large memory, takes time to compute kernel values and the histogram, as a step function, might provide poor modelling of the true, unknown probability density function, with the tails of the probability density function often missing.

### 3.5 MEAN-SHIFT BASED ESTIMATION

Mean-shift vector technique (or) Sequential Kernel Approximation is recently employed for various pattern recognition problems (image segmentation, tracking, etc.). The mean-shift vector is an effective gradient-ascent technique able to detect the main modes of the true pdf directly from the sample data with minimum set of assumptions [11]. Here, the numbers of modes are unrestricted. The mean-shift based method is iterative, thus it is very slow, requires high memory, high cost and requires study of convergence over the whole data space. As such, it is not immediately applicable to modelling background probability density functions at the pixel level. The sample point density estimator with a variant normal kernel, computed at the point $X$ is given by,

$$f\hat{}(x) = n^{-1}(2\Pi)^{-d/2} \sum\sum_{i-1}^{n} |H_i|^{-1/2} \exp\left(-\frac{1}{2} D^2(x, x_i, H_i)\right) (9)$$

where, $D^2(X, X_i, H_i) = (X - X_i)^T H_i^{-1}(X - X_i)$. Here $D^2(.)$ is the Mahalanobis distance from $X$ to $X_i$. Mean-shift vector is used only for an off-line model initialisation [15]. A way to overcome the computational issue is to use this method for only detecting the background PF modes at the initialization time and then use a method that is computationally lighter such as mode propagation. In [16] Piccardi and Jan propose some computational optimisations promising to mitigate the computational drawback. In [15], the authors compared the probability density function obtained with their method against that of a KDE approach over 500-frame test video and thereby finding a low mean integrated squared error in the order 10-4 and justify the name sequential kernel density approximations (SKDA). Sequential Kernel density approximations use mean-shift mode detection only at initialisation time. After the initialisation, the modes are propagated by adapting them with new samples. This method can effectively model a multimodal distribution without the need of assuming the number of modes a priori, faster than KDE and has only low memory requirements.

### 3.6 UNIVERSAL BACKGROUND SUBTRACTION ALGORITHM

Oliver Barnich prescribed to deal with three assumptions in order to be successful in real time applications, 1) What is the model and how does it behave? 2) How is the model initialised? 3) How is the model updated overtime? Normally in statistical method, calculating mean and variance is the main core. The most common method for statistical is Probability Density Function (pdf). It is more reliable to estimate the statistical distribution of background pixel with a small number of close values than with a large number of samples. Each background pixel is modelled by a collection of $N$ background samples values taken in previous frames. $\mu(x) = \{V_1, V_2 \ldots, V_N\}$. Here the Background model is initialized from a single frame. In the response to sudden illumination changes in straight forward, the existing background model is discarded and a new model is initialized instantaneously. How to continuously update the background model with each new frame? This update process must be able to adapt to lighting changes to handle new objects that appear in scene. This universal background model has a choice to update the background model information. There are two different update models such as: conservative and blind update model. In Conservative update scheme, Kernel based pdf estimation technique has a soft approach towards updating. This model never includes a sample belonging to a foreground region in the background model. A pixel sample can be included in the background model only if it has been classified as a back ground sample. In Blind update model, Samples are added to the background model based on whether they have been classified as background or not. A pixel covered by slowly moving objects for more than 10 seconds would still be included in the background model. This update method incorporates three important components: 1) Memory less update policy, which ensures a smooth dealing life span for the samples stored in the background pixel models, 2) A random time sub sampling to extend the time window covered by the background pixel models. 3) A mechanism that propagate background pixel samples partially to ensure spatial consistency and to allow the adaptation of the background pixel models that are masked by the foreground. This Algorithm has detection support map which contains the no. of consecutive times that a pixel has been classified as foreground. The major drawback of this approach is that sometimes old variable value is repeated by new one.

## 4. STRUCTURE OF THE PROPOSED METHOD

In order to separate foreground objects from the background seen image, we need at least two sample images to construct the background in this model. Before going to major steps of the proposed work, we need to incorporate pre-processing. It consists of a collection of simple processing methods that changes the raw input into a format that can be processed by subsequent steps. In the initial stage, let us consider two sample key image of size $m \times n$. Each sample image is organised as row and column vector of background dimension where $k = m$ and $l = n$. In Eq.(1) $fr_{t-1}(a, b)$ the previous key frame with or without foreground objects and $fr_t(a,b)$ is the current frame. Where, $a = 1, 2 \ldots m$ & $b = 1, 2 \ldots n$. The two sample frames in the data matrix can be highly correlated since both contain a large portion of the same stationery background region. Normally for foreground detection we have to calculate difference between previous frame and current frame. In this proposed system, select any 3 key frame (either continuous / or randomly) from first 30 frames in the video sequence. Most of background modeling work, authors has considered 30 frames per second. In this work we are also considering 30 frames, but the major work will be completed in these 30 frames only.

The proposed algorithm is divided into four major steps as shown in Fig.1. According to the frame difference mask of past

several frames, pixels which are not moving for long time are considered as background in the background buffer block. The Background difference mask is generated by comparing the current input and background model image which is stored in the buffer. The background difference mask is one of the major steps involved in this proposed method to generate / detect the foreground moving objects. The object detection is constructed from the background difference mask and frame difference mask. The details of each step will be discussed in the following sub sections.

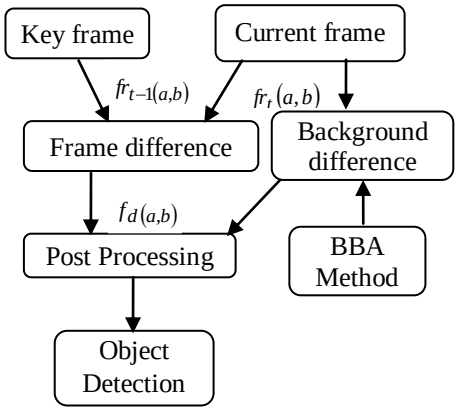



Fig.1. Overview of proposed system

## 4.1 FRAME DIFFERENCE

The background image can be calculated by subtracting the current frame from the previous frame or from the average image of a number of frames. This method works well only in particular conditions such as specific object speed, background noise and frame rate and is very sensitive to the threshold. According to this scheme, pixels belongs to foreground if

$$f_d = \sum |f_r(a,b)| where, B > T \quad (10)$$

where, $T$ is the chosen threshold. The median/average method uses the average or the median of the previous n frames as the background image, $B$. It is quick but very memory consuming. Higher order statistics methods uses threshold to obtain more object shape data by using boundary property of the objects. This information is passed to the background modelling step where the constant background is constructed from the accumulated information of several fame differences

## 4.2 BACKGROUND DIFFERENCE

It is similar to the frame difference method. Background difference mask is done by thresholding the difference between the current frame and the background information which is stored in the background buffer.

$$F_g = |B_m(a,b) - fr_t(a,b)| \quad (11)$$

where, $F_g$ is the background difference and $B_m(a,b)$ is the background model in the buffer and $fr_t(a,b)$ current frame.

## 4.3 BACKGROUND BUFFERING

The background buffering is mainly focused to maintain constant pixel background for modelling the background in order to process the video sequence. This step is used to improve the performance speed in coding level as well as to process uncovered background region. The uncovered background pixel was generated by the complex coding in the existing work. It was proposed by many authors. But in this work, the constant background information is generated by a simple way. The constant background is stored in buffer for succeeding steps in the proposed method. In the storage stage, the history of frame difference mask pixels, background difference mask and constant background pixels are updated in buffer.

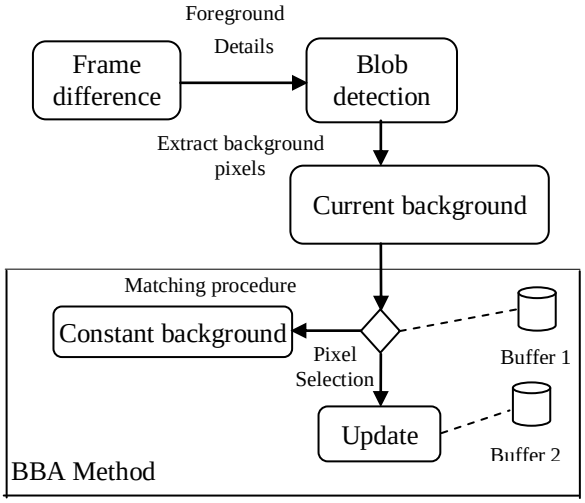



Fig.2. Back ground Buffering process

Initially the foreground object detected by frame difference will provides the information about moving object as well as information about noise. From the frame difference mask, the moving object length and width will be calculated, this length and width of the detection window will continue to further frames. The blob window is simpler to blob detection of tracking method. This will be considered as bounded box over an object in each frame and it will be continued for tracking the particular object or collection of objects state information. Except the blob or rectangular window, all the remaining data will be considered as background. This background pixel information is stored in a buffer and this will be maintained and updated as per the changes in each frame of constant pixels. This blob window always used to predict objects position in each frame. The initial stage of background value is stored in one buffer and remaining updating stage will be stored in other buffer. If any change is found in the background that will update in buffer1 offer exceed the threshold count of the pixel value. It shows that it is a constant stable pixel value of background. This BBA algorithms mostly constant background with less noise. There model can be subtracted to current input frame. Once subtracted, we will get foreground objects, but this will appear to normal frame difference to check whatever we get proper result or not. The morphology and level set method is very useful to get proper foreground without noise values. Processing steps are generally used to eliminate the noise region. In initial stage, the connected components algorithm is applied to mark each isolated region. Then the area of each region is calculated. Region with smaller area than a threshold value are removed from object mask. Foreground may have few noises to degrade the quality of object. There are two kinds of noise, noise in the background and foreground. In the first stage of noise removal, region filter

was applied followed by which morphology operations of close and open are applied, so that it removes all unwanted noise either in background or foreground of each frame.

## 5. PROPOSED SYSTEM FOR OBEJCT DETECTION

The improvement of recommended system is that it does not require any reference frame for performing background modelling and subtraction. This system consumes less time, less memory and provides better result when compared to the existing systems. The proposed system is functions as follows: Followed by segmentation and morphology operations are applied to get reliable result. The final outcome is the foreground object.

### 5.1 BACKGROUND UPDATING

Modified moving average is used to compute the average of frames for initial background model generation. For each pixel (*a,b*) the corresponding value of the current background model *B*(*a,b*) is calculated using the formula

$$B(a,b) = fr_t(a,b) + 1/E\left[fr_t(a,b) - fr_{t-1}(a,b)\right] \quad (12)$$

where, $fr_{t-1}(a, b)$ is the previous background model, $fr_t(a, b)$ is the current incoming video frame, *t* is the frame number. Under the assumption that there is no change in the current pixel, the frame difference obeys zero-mean distribution and its probability density function is shown in the following equation:

$$P\left( fd \middle| H_0 = \frac{1}{\sigma\sqrt{2\pi}} \exp\left( -\frac{fd^2}{2\sigma^2} \right) \right) \quad (13)$$

where, *fd* is the frame difference and $\sigma^2$ is the variance of the frame difference and is equal to twice the noise variance. $H_0$ is null hypothesis, i.e., there are no changes at the current pixel. While detecting the blob window, we need to know the state of objects. The object state can be expressed as follows,

$$X_i = PX_{i-1} + QC_{i-1} + R_{i-1} \quad (14)$$

where, *C* is the optional control input, *N* is the process noise or unknown zero mean white Gaussian noise, $X_i$ is the state vector. This is similar to predict-correct-update cycle of kalman filter for tracking the objects. The blob detections will continue for each and every frame with the help of state estimation process. For the next frame state, we consider only posterior error to correct the state information for the specified object. The posterior error $e_k^-$ is given as follows,

$$e_k^- = X_{i+1} - X_i \quad (15)$$

Background Buffering is done in order to construct an optimal background model.

### 5.2 PIXEL SELECTION/OPTIMIZATION

It is used to extract stable pixel in the incoming video sequence. 1) Determination of background pixel via matching procedures. 2) Use of stable pixel in order to provide a measure of temporal activity of the pixels within the set of back ground pixel. 3) Determination of the optimum background pixel via accurate matching procedure. The pixel selection procedure has fixed record. If a pixel is marked as changing in the frame difference mask, the corresponding value in the fixed record is cleared as zero, otherwise, if the pixel is fixed, the corresponding value is incremented by one. The matching procedure is used to quickly find a great quantity of background candidate by determining whether or not their respective pixel values for the incoming video frame. If (*a*, *b*) are equal to the corresponding pixel value of the previous frame.

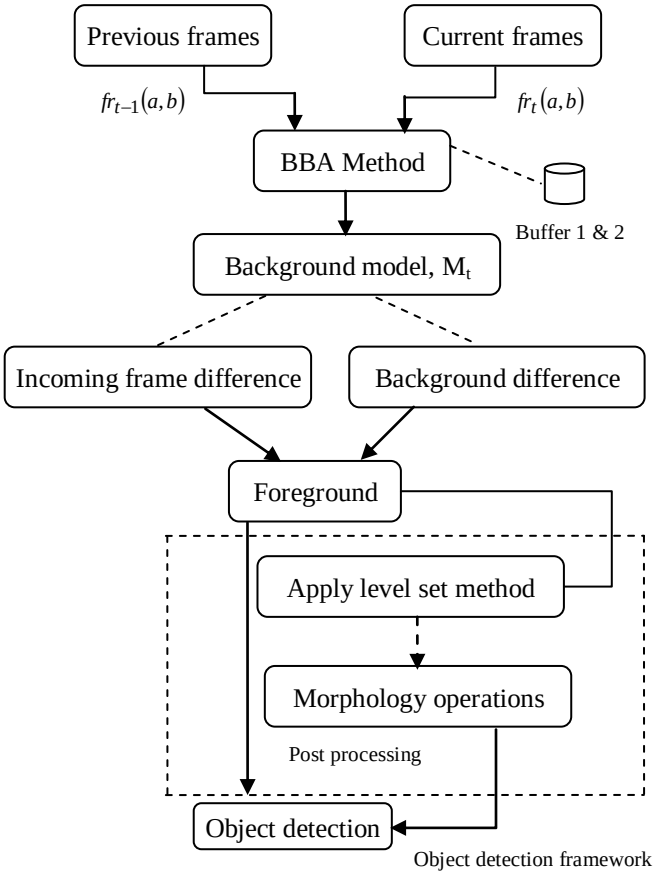



Fig.3. Overview of object detection framework

### 5.3 BACKGROUND BUFFERING APPROACH

Each optimum background pixel of $M_t(x,y)$ will then be supplied to every frame of the background model *B*. The optimum background model updated as follows: If a pixel is marked as changing in the frame difference mask, the corresponding value in the fixed record is cleared as zero, otherwise, if the pixel is fixed, the corresponding value is incremented by one. Based on optimum background model the best possible background pixels are then updated for the background model. This will give the best results when detection is performed. The moving average formula used in Eq.(12) has a predefined parameter, *E*. If the pixel is constant for past several frames, then there is a high probability that it belongs to the background region.

***Algorithm:***

**Step 1**: For $i = 1$ to 30 frames

$f_d = fr_{i-1} - fr_i$

// $f_d$ is the frame difference with respect to previous frame. Repeat the above steps for random frame.

**Step 2**: Fixed blob around the object using state representation.

**Step 3**: Except blob the Extract background information for buffer storage.

**Step 4**: For each and every background $B_m$ is compared with previous background in buffer $B_{mt\text{-}1}$

a. Check whether $\delta B_m > T$, then it is considered for background updation. Otherwise it is not needed to update the value.

b. If $B_{mt\text{-}1}$! = $B_{mt}$ then Update $\delta B_m$. Otherwise keep the existing buffer values.

**Step 5**: New background buffer will be changed.

$$B_m = \sum \delta B_m + B_{mt}$$

**Step 6**: Foreground detection subtract new frame with background frame from the buffer.

**Step 7**: To apply level set and morphing, easily remove the noise from the foreground.

## 6. EXPERIMENTAL ANALYSIS

We have implemented the proposed background modelling method using Matlab on a computer with 2 GB RAM and 2.1 GHz Intel 3 processor. This processor capture video using video capture card, AVI format, the frame rate is 30fps and the size of the image is 180 × 144 pixels. This proposed method is tested on various image sequences like Weizmann dataset run part 1 and Weizmann dataset walk Part 2. In this experimental analysis, detection window is selected as per the direction of the object movement. Select the three or five key frames randomly mostly from the first 30 frames in the video sequence. These frames are converted into images and thereby to gray scale in order to overcome computation complexity. The width and height of the moving object is calculated from the key frames. The various backgrounds are obtained from those key frames with the help of blob window. For the buffering of background process involves the following steps: the initial stage of key frame background will be stored in buffer.

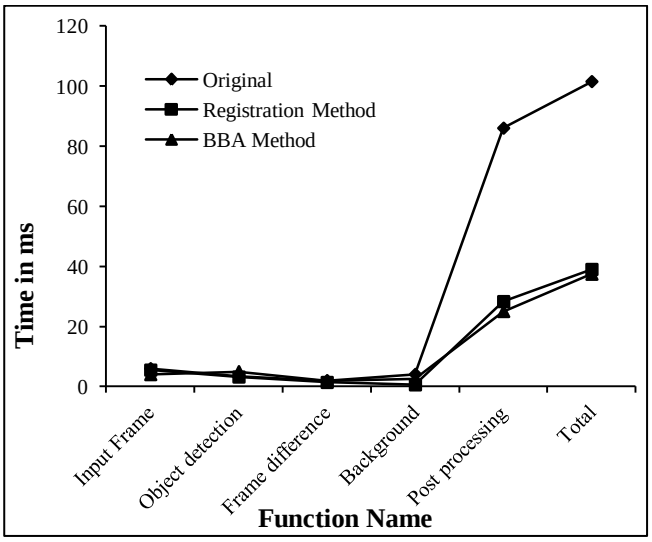



Fig.4. Comparison chart for Execution Time analysis of BBA

After that, the incoming background frames are compared with existing background information. If any changes found in the background pixel will be checked whether the pixel belongs to constant or dynamic. If it is a dynamic state that will be stored in buffer2 for current processing background. If it is a constant pixel then there are no changes in the buffer1. As per the background storage we have to maintain two buffers for this background modelling. If changes are detected, then background registration process is performed. The registration is done recursively for first 30 frames. Now the background is retrieved from the background registration process.

This background is used as a reference background and background subtraction is performed. One of the drawbacks that confuse the conventional change detector is that, the object may stop moving temporarily or move very slowly. When comparing the analysis of proposed method (Table.1 and Fig.4) to the existing, it shows an improvement in the run time value.

Table.1. Execution time analysis of BBA

| Function name | Original | Registration Method | BBA Method |
|---|---|---|---|
| **Input Frame** | 6 | 5.5 | 4 |
| **Object detection** | 3.4 | 3.2 | 5 |
| **Frame difference** | 2 | 1.4 | 1.9 |
| **Background** | 4.1 | 0.6 | 2.5 |
| **Post processing** | 86 | 28.3 | 25 |
| **Total** | 101.5 ms | 39 ms | 37.4 ms |

In general, Running Gaussian Average method needs high memory requirements compared to Mixture of Gaussians. Other methods like frame difference and MOG needs high memory requirements and shows some intermediate results. But the proposed BBA method improves the processing speed, needs only less memory, as well as good accuracy of the results (Fig.5(a) & Fig.5(b)) is obtained. On the other hand, the motion estimation is not very accurate near the object boundary where highest accuracy is required. Motion estimation can deal with the translation type of motion, only if the other forms of movement are involved, otherwise motion vector may fail to track the object motion. The comparison execution time for different method is presented in Fig.4. In registration method, [11] has proven to reduce processing time from 101.5ms to 39.ms. BBA algorithm has reduce to 37ms (almost 64.1ms) as well as it need only less memory space for buffering intermediate results. The error rate of object mask is adopted to present the effectiveness of proposed algorithm. The error rate can be expressed as the following equation

$$error\ rate = \frac{error\ \ pixel\ count}{frame\ size} \tag{16}$$

The error rate of BBA technique is lower than 0.7% most of the time, with an exception that a sudden rise of error started at initial frame. After the background information is buffered in the background region, the error rate comes down to normal value.

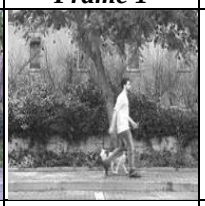
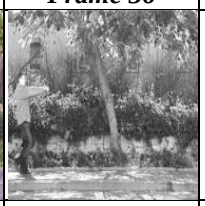
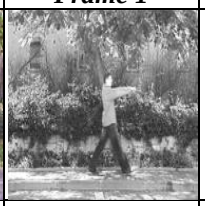
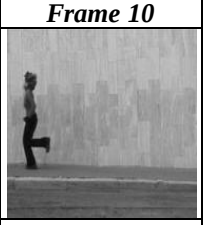
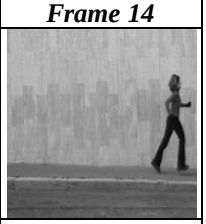
| Data Set | Frame Sequence | | RGB to Gray frame | |
|---|---|---|---|---|
| Walking in the trees |  |  |  |  |
| | *Frame 1* | *Frame 18* | *Frame 1* | *Frame 18* |
| |  |  |  |  |
| | *Frame 36* | *Frame 54* | *Frame 36* | *Frame 54* |
| |  |  |  |  |
| | *Frame 1* | *Frame 20* | *Frame 1* | *Frame 20* |
| |  |  |  |  |
| | *Frame 40* | *Frame 66* | *Frame 40* | *Frame 66* |
| Running |  |  |  |  |
| | *Frame 1* | *Frame 10* | *Frame 1* | *Frame 10* |
| |  |  |  |  |
| | *Frame 20* | *Frame 30* | *Frame 20* | *Frame 30* |
| |  |  |  |  |
| | *Frame 1* | *Frame 14* | *Frame 1* | *Frame 14* |
| |  |  |  |  |
| | *Frame 21* | *Frame 28* | *Frame 21* | *Frame 28* |

Fig.5(a). Weizmann Data set (Walking in Tree, Running)

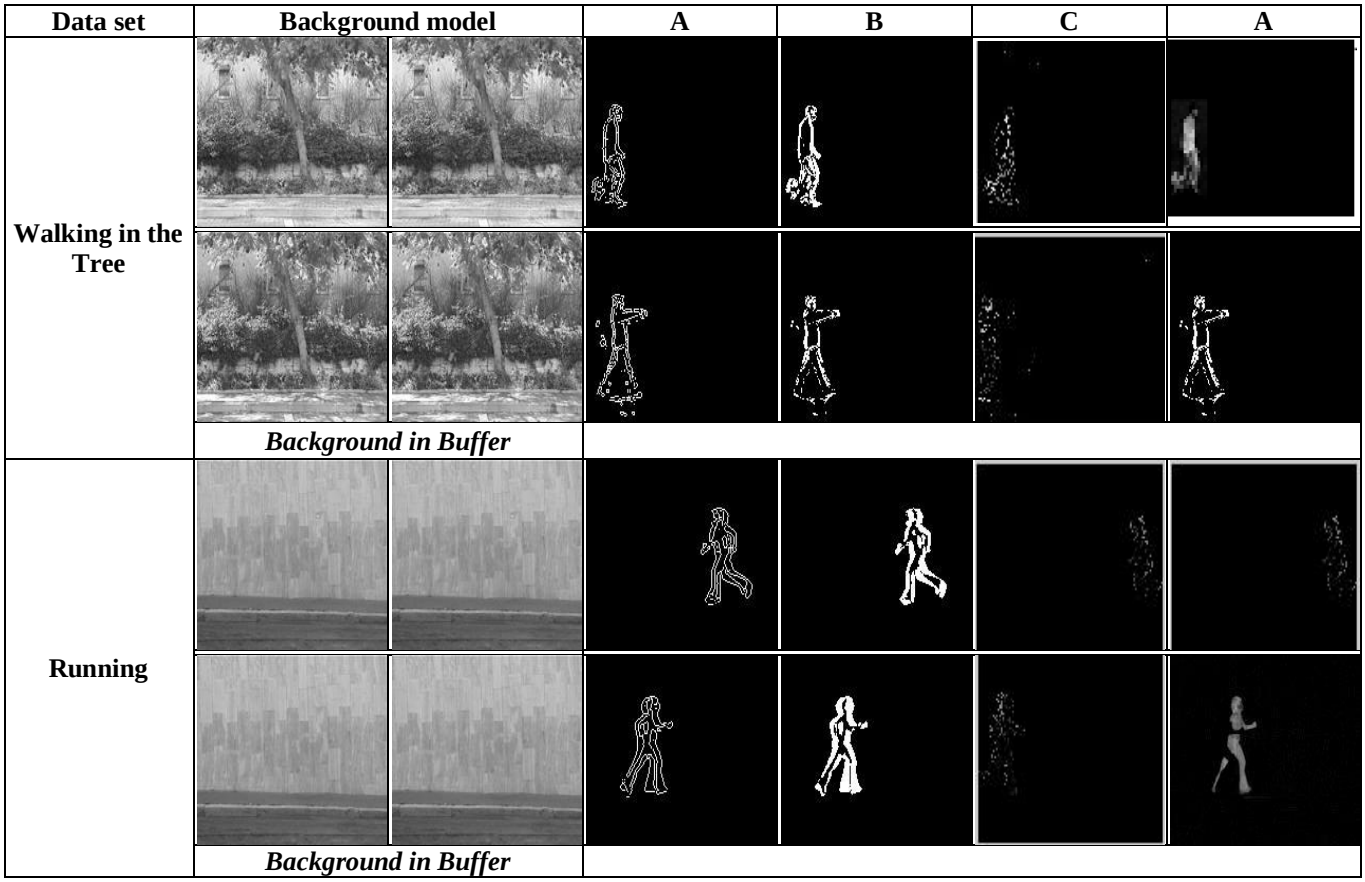



Fig.5(b). Foreground in Frame difference: A) Foreground in BS, B) Foreground in FD, C) Foreground in MOG, D) Foreground in Buffering

## 7. CONCLUSION

An efficient background modelling with Background Buffering Algorithm was formulated and implemented to construct a reliable background pixel from accumulated background difference and frame difference. Each frame pixel is compared to buffer. Except moving object data other data is used to construct reliable background information. Once the background model is constructed, then a post processing step is applied on the obtained object mask to remove noise region and to smoothen the object boundary. When running personal computer with 2 GB RAM and 2.1 GHz Intel 3 processor, the experimental results demonstrate that good modelling and segmentation with better quality are obtained. Therefore this algorithm is very useful for the real-time surveillance system.

## ACKNOWLEDGEMENT

I am thankful to my Professor Dhananjay Kumar for his guidance, suggestions, and help and to structure this paper.

## REFERENCES

[1] Meghna Singh, Anup Basu and Mrinal Kr. Mandal, "Human Activity Recognition Based on Silhouette Directionality", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, No. 9, pp. 1280-1292, 2008.

[2] Carlos R. del-Blanco, Fernando Jaureguizar and Narciso Garcıa, "Bayesian visual surveillance: a model for detecting and tracking a variable number of moving objects", *18th IEEE International Conference on Image Processing*, pp. 1437-1440, 2011.

[3] Eduardo Monari and Charlotte Pasqual, "Fusion of background estimation approaches for motion detection in non-static backgrounds", *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp.347-352, 2007.

[4] M. Murshed, A. Ramirez and O. Chae, "Statistical Background Modelling: An Edge Segment Based Moving Object Detection Approach", *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 300-306, 2010.

[5] O. Barnich and M. Van Droogenbroeck, "ViBe: A Universal Background Subtraction Algorithm for Video Sequences", *IEEE Transactions on Image Processing*, Vol. 20, No. 6, pp. 1709-1724, 2011.

[6] Lun Zhang, S.Z. Li, Xiaotong Yuan and Shiming Xiang, "Real-time Object Classification in Video Surveillance Based on Appearance Learning", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[7] R.N. Hota, V. Venkoparao and A. Rajagopal, "Shape Based Object Classification for Automated Video

Surveillance with Feature Selection", *10th International Conference on Information Technology*, pp. 97-99, 2007.

[8] Hong Lu, Hong Sheng Li, Lin Chai, Shu Min Fei and Guang Yun Liu, "Multi-Feature Fusion Based Object Detecting and Tracking", *Journal on Applied Mechanics and Materials*, Vol. 117-119, pp. 1824-1828, 2011.

[9] C. Beyan and A. Temizel, "Adaptive mean-shift for automated multi object tracking", *IET on Computer Vision*, Vol. 6, No. 1, pp. 1-12, 2012.

[10] Blanco Adán, Carlos Roberto del and Jaureguizar Nuñez, Fernando and García Santos, "Bayesian Visual Surveillance, a Model for Detecting and Tracking a variable number of moving objects", *IEEE International Conference on Image Processing*, pp. 1437-1440, 2011.

[11] Massimo Piccardi, "Background subtraction techniques: a review", *IEEE International Conference on Systems, Man and Cybernetic*, Vol. 4, pp. 3099-3104, 2004.

[12] Ahmed Elgammal, David Harwood and Larry Davis, "Non-parametric Model for Background Subtraction", *Computer Vision - Lecture Notes in Computer Science*, Vol. 1843, pp. 751-767, 2000.

[13] T. Bouwmans, F. El Baf and B. Vachon "Background Modelling using Mixture of Gaussians for Foreground Detection - A survey", *Recent Patterns on Computer Science 1*, pp. 219-237, 2008.

[14] Donovan H. Parks and Sidney S. Fels, "Evaluation of Background Subtraction Algorithms with Post-processing", *Fifth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 192-199, 2008.

[15] M. Hedayati, Wan Mimi Diyana Wan Zaki and Aini Hussain, "Real-Time Background Subtraction for Video Surveillance: From Research to Reality", *6th IEEE International Colloquium on Signal Processing and Its Applications*, pp. 1-6, 2010.

[16] M. Piccurdi and T. Jan, "Mean-shift Background image modelling", *IEEE International Conference on Image Processing*, Vol. 5, pp. 3399-3402, 2004

[17] C. Wren, A. Azarhayejani, T. Darrell and A.P. Pentland, "Pfinder: real-time tracking of the human body", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 780-785, 1997.

[18] Chris Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 246-252, 1999.

[19] Ali Harimi and Alireza Ahmadyfard, "Image segmentation Using Correlative Histogram Modeled by Gaussian Mixture", *IEEE International Conference on Digital Image Processing*, pp. 397-401, 2009.