

BBBCO and fuzzy entropy based modified background subtraction algorithm for object detection in videos

Manisha Kaushal¹ · Baljit Singh Khehra²

Published online: 8 May 2017
© Springer Science+Business Media New York 2017

Abstract Background subtraction (BS) is one of the most commonly used methods for detecting moving objects in videos. In this task, moving object pixels are extracted by subtracting the current frame from a background frame. The obtained difference is compared against a threshold value to classify pixels as belonging to the foreground or background regions. The threshold plays a crucial role in this categorization and can impact the accuracy and preciseness of the object boundaries obtained by the BS algorithm. This paper proposes an approach for enhancing and optimizing the performance of the standard BS algorithm. This approach uses the concept of fuzzy 2-partition entropy and Big Bang–Big Crunch Optimization (BBBCO). BBBCO is a recently proposed evolutionary optimization approach for providing solutions to problems operating on multiple variables within prescribed constraints. BBBCO enhances the standard BS algorithm by framing the problem of parameter detection for BS as an optimization problem, which is solved using the concept of fuzzy 2-partition entropy. The proposed method is evaluated using videos from benchmark datasets and a number of statistical metrics. The method is

also compared with standard BS and another recently proposed method. The results show the promise of the proposed method.

Keywords Videos · Background subtraction · BBBCO · Object detection and tracking · Threshold

1 Introduction

In computer vision and image processing, moving object detection and tracking plays an important role in higher-level video analysis applications. It is the first and foremost step in various applications such as intelligent traffic surveillance [1, 2], real-time detection of suspicious behavior in shopping malls [3], path detection in video surveillance [4], and intelligent observation of birds for further use in various applications such as aviation safety and protection [5]. Object detection can be done by extracting the shape of a moving object from a video sequence using a number of techniques like background subtraction (BS), feature extraction, and statistical methods. Object detection and tracking go hand in hand; once an object is detected, it is tracked throughout the video sequence using different techniques. There are two major activities involved in all object detection methods, namely background construction and foreground extraction. In background construction, the background information is constructed using different methods such as taking the arithmetic mean of pixels between successive frames, statistical methods, or hybrid methods. The constructed background is later subtracted from successive frames to obtain the desired object. Foreground extraction-based object detection uses temporal, spatial, or spatiotemporal information to extract an initial video object and later utilizes motion-based information to detect objects

✉ Manisha Kaushal
er.manishakaushal@gmail.com

Baljit Singh Khehra
baljit.singh@bbsbec.ac.in

¹ UIET, Panjab University Chandigarh,
Chandigarh, 160014, India

² Department of CSE, BBSBEC Fatehgarh Sahib,
Punjab 140407, India

in successive frames. Because various issues such as illumination changes, fake motion, crowded video scenes, and Gaussian noise are involved, it is challenging to perfectly detect and track objects in a dynamic and practical environment. The performance of detection systems can be drastically affected by small movements within the environment. The challenges imposed by the abovementioned issues make the problem of detecting and tracking moving objects from videos a very interesting, challenging, and multidisciplinary research problem in computer vision and image processing. Furthermore, it is desirable to devise a mechanism for achieving more accuracy in the detection and tracking objects with as few misses as possible.

This paper is organized as follows: Section 2 summarizes various studies for background detection available in the literature. Section 3 details about various methods used in this study like Big Bang Big Crunch Optimization (BBBCO), fuzzy 2-partition entropy. Section 4 elaborates the proposed approach; algorithms used in the proposed modified BBBCO based BS are described in detail. Experiments and results are presented in Section 5, followed by conclusions in Section 6.

2 Related work

Background subtraction approaches consist of finding the absolute difference between current frame and background so that relevant changes could be detected. The success of algorithm lies in devising an effective method on the basis of which background can be modeled and updated. Background subtraction performs well in extracting objects from videos but the effectiveness of approach degrades as the background is not static, has lighting fluctuations and videos are noisy. A number of methods have been proposed for improvement of standard BS algorithm but these have been targeted for specific applications or environments. It is also difficult to devise any general method that is accurate in all situations. Various BS methods differ in mechanism by which they perform modeling of background and metrics they use for classification. The following section summarizes various studies available in the literature.

Heikkila et al. [6] used the conventional BS approach to identify moving objects in videos. The study used a first-order recursive filter to update the background model for reliable moving object detection. Mandellos et al. [7] presented a background reconstruction algorithm that uses a histogram filtering procedure. The most common algorithm used in BS models is based on Gaussian mixture models (GMMs). Yoshinaga, in [8] presented a GMM spatiotemporal BS model that performs a region level statistical analysis with a selection of nine sensitive parameters. These parameters were continuously updated as the video was

analyzed. Chen and Ellis [9] proposed a self adaptive GMM model with an adaptive parameter, median of quotient, which implements special filters to suppress image noise and sudden illumination changes. The model was able to handle shadows and reflection issues, and a final morphological operation was incorporated to fill holes in the final frame. However, the detections were not very accurate. Spampinato et al. [10] proposed a texton-based kernel density estimation that used color and texture features. The texton-based method was validated using three different databases: I2R [<http://perception.i2r.a-star.edu.sg/bkmodel/bkindex.html>], [Fish4Knowledge <http://groups.inf.ed.ac.uk/f4k/F4KDATASAMPLES/INTERFACE/DATASAMPLES/search.ph>], and Background Modeling Challenge (BMC) <http://bmc.iut-auvergne.com/>. The results indicated problems with night videos, scarce illumination, highly dynamic backgrounds, and rain/snow scenarios. Comprehensive surveys of the state-of-art techniques for conventional and recent approaches to background modeling and foreground detection were presented by Bouwmans [11], Shaikh et al. [12], Piccardi [13], Sobral et al. [14], and Karasulu et al. [15].

Entropy-based detection methods for videos are rather sparse in the literature, and a few of them are discussed in this section. Qin et al. [16] proposed a layered segmentation approach for detecting stationary and moving objects from surveillance video. Their method extended the maximum entropy statistical model to enable it to segment layers with certain features that were collected by developing a codebook with a set of codewords for each pixel. Wang et al. [17] proposed a block-based approach that utilizes entropy information collected in an H.264-bit stream for extracting moving objects. They applied a block entropy feature from compressed domain video to handle bit streams with variable quantization values. Two background entropy models were built for each video block to reflect the difference in coding efficiency between inter- and intra-block types. The foreground frame was built using BS after a normalization process. Over-segmentation when extracting a moving object was avoided by applying a hysteresis thresholding approach. Morphological filters and connected component labeling were applied at the final stage to extract the list of moving objects. Badri et al. [18] calculated the threshold value for difference images using an adaptive thresholding algorithm whose window size was chosen by the entropy measure over the window under consideration. An entropy-based window growing approach was used in which the ideal size of the window was generated by comparing the entropy of the window with the image histogram and using Ostu's thresholding method for binarization. The results were poor and no benchmark dataset was utilized for the evaluation. Ma and Zhang [19] proposed a spatio-temporal entropy-based object detection method

by measuring the color variation in successive frames and building a spatiotemporal entropy image. They claim that the proposed method is more robust to noise than the traditional difference-based methods. However, the proposed method generates false detections when light flickering and shadows exist in the scene. The most relevant study similar to the present work was proposed by Karasulu and Korukoglu [20]. They implemented a modified BS method based on spatial entropy and formulated the threshold detection problem as a p-median problem. Demand and facility points for the p-median problem were used as the entropy for the foreground and background regions. A modified BS algorithm based on simulated annealing (SA) was also implemented.

The literature discussed in this section summarizes the various research directions in which the standard BS algorithm has been explored. A good BS algorithm requires accurate pixel classification, should be robust, and must be adaptive to environment changes. Fortunately, such challenges have been successfully handled using soft computing techniques. Some solutions based on soft computing approaches have been reported in the literature to deal with noisy and insufficient data, but their applicability on real and noisy videos is still an open issue. Designing an effective system requires a soft computing approach that can optimize the detection process in a real dynamic environment to improve accuracy.

3 Problem analysis and basic methods

This section briefly explains Big Bang Big Crunch Optimization (BBBCO), entropy method, fuzzy 2- partition entropy and details of how to effectively determine threshold value using the concept of BBBCO method.

3.1 Big bang big crunch optimization

In recent times, meta-heuristic algorithms such as Genetic Algorithms, Particle Swarn Optimization algorithms, Ant-Colony Optimization algorithm etc. have been successfully proven to propose solutions for a wide variety

of applications. The Big Bang–Big Crunch Optimization (BBBCO) algorithm is inspired by theory for origin and evolution of universe consisting of two phases: big bang and big crunch. In Big Bang phase, energy dissipation produces a disorderly state of particles, whereas, in Big Crunch phase, the randomly distributed particles are drawn into an order. Randomness is the key feature of the Big Bang phase. Randomness resembles energy dissipation in nature while convergence to a local or global optimum point can be seen as a gravitational attraction. Since energy dissipation creates disorder from ordered particles, randomness is used as a transformation from a converged solution (order) to the birth of totally new candidate solutions (disorder). Figure 1 summarizes the steps of standard BBBCO algorithm. This optimization is based on the application of two sequential phases: Big Bang and Big Crunch. Big Bang phase consists of initializing the population of candidate solutions randomly over the entire search space like other evolutionary search algorithms. In the next phase, Big Crunch, the contraction operator calculates the center of mass by processing current position of each candidate solution and its associated fitness value. The Explosion (Big Bang phase) and contraction phase (Big Crunch phase) are repeated until specified criteria are met. BBBCO is performed with an aim to shrink the randomly generated solutions to a single point using center of mass computations as the base. The calculation of center of mass is done using formula given in (1)

$$C = \frac{\sum_{i=1}^N \frac{x_i}{F_i}}{\sum_{i=1}^N \frac{1}{F_i}} \quad (1)$$

where C is the position of the center of mass; x_i is the position of candidate i in an n -dimensional search space; F_i is the fitness value of the candidate i ; and N is the population size in Big Bang phase. The algorithm proceeds to calculate new candidate solutions in and around this center of mass using formula (2)

$$x_{\text{new}}^{k+1} = x_c^k \pm \frac{l \cdot r}{k} \quad (2)$$

where x_c^k is the center of mass in k^{th} iteration; x_{new}^{k+1} is the new center of mass in $(k+1)^{\text{th}}$ iteration; r is a random

Fig. 1 Basic Big Bang Big Crunch optimization(BBBCO) algorithm

Step1: Initial Big Bang Phase	Generate initial population randomly overall search space.
Step 2: Evaluate Fitness Value	Compute fitness value of candidate solutions using defined formula.
Step 3: Big Crunch Phase	Compute center of mass for the current population of candidate solution using (1).
Step 4: Big Bang Phase	Determine new candidate solution around the center of mass, computed in step 2 using formula defined in (2).
Step 5: Stopping Criteria	Return to step 2 until the stopping criteria are met.

number; l is the upper limit on parameter and k is the current iteration step.

The aim of standard BBBCO is to find an appropriate solution for which value of fitness function is minimized. The major advantage of BBBCO lies in its ability to avoid trapping in local minima because of the random search strategy employed by it. BBBCO method is also computationally less expensive and provides quick convergence as required for real time processing applications. The readers may refer to [21–23] for more details.

3.2 Fuzzy 2-partition entropy approach

It is generally believed that image processing contains some fuzziness in nature and the fuzzy entropy method is a well-known threshold detection approach in image processing. Several researchers have utilized the concepts of image entropy for extracting suitable threshold values for different image sequences. In this paper, we use the concepts of a fuzzy 2-partition and maximum fuzzy entropy principle to select an optimal threshold values for performing background subtraction in videos.

Consider any digital image I of size $M \times N$ and let $f(x,y)$ be gray level value of pixel (x,y) ranging from $(0,1,2,\dots,L-1)$. If the number of pixels with gray level k is denoted as $h(k)$, then, the probability of occurrence of the gray level k in the image can be calculated as:

$$p(k) = h(k) / MxN \tag{3}$$

To partition the image into foreground and background, two different formulas for calculating fuzzy membership values μ_F and μ_B been used. The foreground region can be modeled using μ_F and the background region can be

modeled using μ_B . The fuzzification of an image pixel into any of these regions can be effectively modeled by membership curves for μ_F and μ_B calculated using (4) and (5), respectively.

$$\mu_F(j) = \begin{cases} 0 & j \leq a \\ \frac{(j-a)^2}{(c-a)(b-a)} & a < j \leq b \\ 1 - \frac{(j-c)^2}{(c-a)(c-b)} & b < j \leq c \\ 1 & j > c \end{cases} \tag{4}$$

$$\mu_B(j) = 1 - \mu_F(j) \tag{5}$$

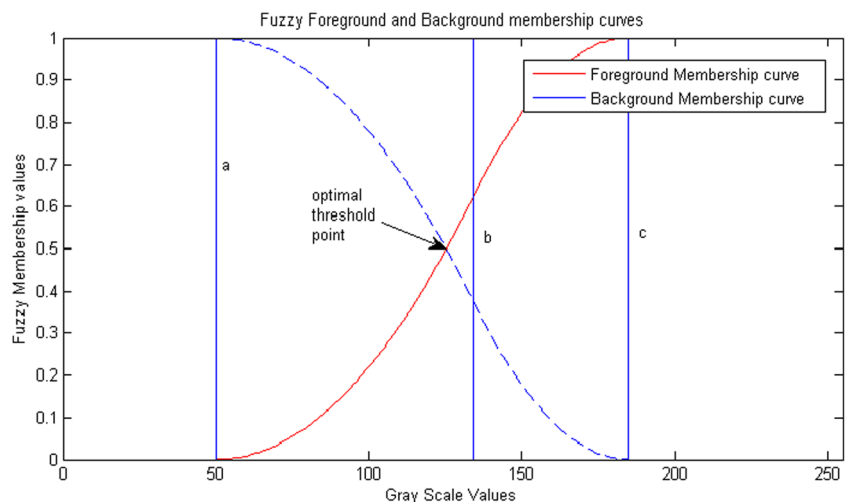
where $0 \leq a \leq L-1$; $0 \leq b \leq L-1$; $0 \leq c \leq L-1$; and $j = 0, 1, 2, 3, 4, \dots, 255$.

Figure 2 shows shapes of these membership functions for sample values of (a, b, c) . The optimal threshold value for classifying any pixel as foreground or background can be extracted from the intersection of the curves μ_F and μ_B , as shown in Fig. 2. It can be clearly seen that $\mu_F = 1 - \mu_B$ and, according to [28], μ_F and μ_B is the fuzzy 2-partition of the image. The two probability distributions for foreground and background can be found using the fuzzy distribution of the gray-level values of the frame. The probability distribution functions for foreground and background is given by formulas defined in (6) and (7).

$$P_F = \sum_{j=0}^{L-1} \mu_F(j) p(j) \tag{6}$$

$$P_B = \sum_{j=0}^{L-1} \mu_B(j) p(j) \tag{7}$$

Fig. 2 μ_F and μ_B membership curves with sample values of (a, b, c)



Chang et al. [28] specified that total entropy for such an image can be calculated using probabilities of fuzzy events of different fuzzy partitions defined as:

$$H = - \sum_{i=1}^c P(A_i) \log_2(P(A_i)) \quad (8)$$

where $P(A_i)$ refers to probabilities of fuzzy events for various partitions, in this case, $c=2$ for background and foreground. H is the total entropy of image or frame. In other words, H may be rewritten as:

$$H = -P_F \log_2(P_F) - P_B \log_2(P_B) \quad (9)$$

$$H = H_F + H_B \quad (10)$$

where $P_F \log_2(P_F)$ and $-P_B \log_2(P_B)$ are entropies of foreground and background regions respectively, also referred to as H_F and H_B for brevity.

Using (6) and (7), we can calculate H_F and H_B as follows:

$$H_F = - \left[\left(\sum_{j=0}^{L-1} \mu_F(j) p(j) \right) \log_2 \left(\sum_{j=0}^{L-1} \mu_F(j) p(j) \right) \right] \quad (11)$$

$$H_B = - \left[\left(\sum_{j=0}^{L-1} \mu_B(j) p(j) \right) \log_2 \left(\sum_{j=0}^{L-1} \mu_B(j) p(j) \right) \right] \quad (12)$$

According to information and statistical theory concepts, the largest value of entropy leads to better solutions. Because the total entropy is the combination of foreground and background regions, the value of entropy is governed by the corresponding curves of μ_F and μ_B , which are defined for the foreground and background, respectively. Curves μ_F and μ_B are contradictory in nature, as an increase in the entropy of the foreground will tend to decrease the entropy of the background. Because the total entropy is a combination of both types of entropies, they will eventually settle into an optimal combination. The intersecting point of the optimal curves for μ_F and μ_B corresponds to threshold values that can be used for classification (see Fig. 2). According to (4) and (5), μ_F and μ_B can be modeled by triplet (a, b, c) ; different (a, b, c) will generate different fuzzy entropy values for the foreground and background pixels. Therefore, the problem is reduced to finding an optimal combination of values of a , b , and c , such that the total entropy of both the foreground and the background pixels is maximum. The values

of the parameters for which value of total entropy (H) is maximum will be used to determine the optimal threshold value for separation of foreground from background using (13) as

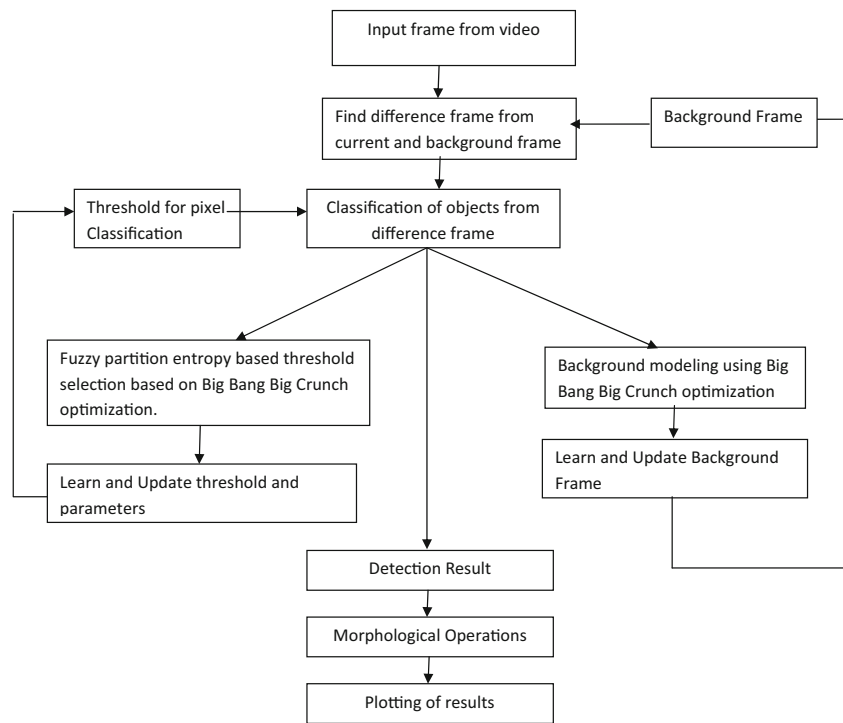
$$(a^*, b^*, c^*) = \arg \max \{H(a, b, c)\} \text{ such that } abc \in L \begin{cases} 0 \leq a \leq (L-1) \\ 0 \leq b \leq (L-1) \\ 0 \leq c \leq (L-1) \end{cases} \text{ and } 0 \leq a \leq b \leq c \quad (13)$$

4 Proposed approach

This paper proposes an automatic and fast approach for the detection of moving objects in videos using an updated BS algorithm. The BS algorithm detects moving objects by calculating the difference of the current frame from a background frame and performing binarization over that difference using a threshold value. The background is continuously updated at every frame to absorb dynamics of the environment. The background update mechanism is a crucial component in the BS algorithm, as effective background modeling helps push static or motionless objects into the new background. The threshold also plays an important role in the classification of pixels to either foreground or background regions. It also helps to suppress false detections and nullify the presence of holes in frames. In this paper, the standard background detection algorithm has been changed and an optimized threshold value is generated using the BBBCO algorithm, which aids pixel classification and the detection of true object boundaries. BBBCO has also been utilized to extract other parameters for the standard BS algorithm, which has paved the way for the effective and correct modeling of background. Figure 3 shows the methodology used in the proposed approach. The motivation for employing BBBCO lies in its advantages over other algorithms, as it is computationally less expensive and converges faster. This makes it suitable for real time solutions and use in low-end devices like video and CCD cameras. In addition, calculation of the center of mass at every step incorporates more random behavior into the population space and helps the algorithm avoid local minima/maxima.

The proposed BBBCO-based method generates a threshold value using the concept of fuzzy 2-partition entropy and formulates the threshold detection problem as a parameter optimization problem. The parameter estimation problem is an optimization problem in which the goal is to search for an optimal combination of parameters, in this case (a, b, c) , within prescribed constraints. The optimization problem can be formulated as follows: find an optimal combination of parameters (a, b, c) such that the total entropy of the

Fig. 3 BBBCO based modified background subtraction methodology



frame is maximized. The objective function can be specified as:

$$H = - \left[\left(\sum_{j=0}^{L-1} \mu_F(j) p(j) \right) \log_2 \left(\sum_{j=0}^{L-1} \mu_F(j) p(j) \right) \right] - \left[\left(\sum_{j=0}^{L-1} \mu_B(j) p(j) \right) \log_2 \left(\sum_{j=0}^{L-1} \mu_B(j) p(j) \right) \right] \quad (14)$$

The aim is to maximize above objective function subject to constraints specified in (15) as

$$\begin{cases} 0 \leq a \leq 255 \\ 0 \leq b \leq 255 \text{ and } 0 \leq a \leq b \leq c \\ 0 \leq c \leq 255 \end{cases} \quad (15)$$

For grey level videos, the search space for (a, b, c) varies in the range (0, 255) and therefore, these values act as constraints for the variables. Once an optimal combination of values for (a,b,c) is found, an optimal threshold value can be found by solving (4) and (5) [28, 29]. The resulting formula is specified in (16)

$$T = \begin{cases} a^* + \sqrt{\frac{(c^*-a^*)(b^*-a^*)}{2}} , & (a^* + c^*)/2 \leq b^* \leq c^* \\ c^* - \sqrt{\frac{(c^*-a^*)(c^*-b^*)}{2}} , & a^* \leq b^* \leq (a^* + c^*)/2 \end{cases} \quad (16)$$

where a^*b^* and c^* are values for which (14) is maximized.

4.1 BBBCO based fuzzy 2-partition entropy thresholding algorithm

The paper uses the BBBCO-based fuzzy 2-partition entropy thresholding approach for finding an optimal value for the threshold and utilizing this value to effectively model a real-time solution for the detection of moving objects in videos. The contribution of this work is novel, as a standard BS method has been optimized and made adaptive using the Big Bang–Big Crunch technique for the effective detection of objects in videos. The methodology makes use of the concept of fuzzy 2-partition entropy and uses BBBCO to threshold videos into two regions, foreground and background. BBBCO is also used to determine parameters that help model the background effectively. The complete steps for modified BS method using BBBCO are given below:

Steps of BBBCO-BS Procedure

1. Read a video sequence; Extract frame information and other video parameters like number of frames (NFrames), frames per second (fps), FrameHeight, FrameWidth.
2. Repeat for $i = 1$ to NFrames /*Run the loop for all frames*/
3. Read new frame as $F(i)$ /* Read new frame*/
4. **Initialization:** Initialize variables
 - 4.1. if (first frame) /* Initialize variables to be used in algorithm for first time*/

- 4.2. Initialize Background (frame) and Foreground (frame) to $F(i)$.
 - 4.3. Use first frame to initialize some matrices that will be used in background subtraction algorithm namely `Difference_matrix`, `Foreground_matrix`, `Background_matrix` /* `Difference_matrix = Foreground_matrix - Background_matrix = first frame` */
 - 4.4. end if.
 5. Generate an optimized threshold value along with other parameters for current frame.
 - 5.1. `if((mod (fps, i))==0)` /* condition to call BBBCO procedure; This is implemented to restrict calling on every frame so that performance can be improved */
 - 5.2. Find `threshold = BBBCO(F(i))`; /* get threshold and other parameter for frame */
 - 5.3. end if
 6. **Evaluation:** Perform evaluation and update background from extracted parameters
 - 6.1. Read next frame as `current_frame`.
 - 6.2. Update `Foreground(frame)` and `Background(frame)` to `current_frame`.
 - 6.2.1. `if (mod(track_refreshrate,i)==0)` /* use value of variable `track_refreshrate` to check if background refresh is required. Lower value means frequent update of background */
 - 6.2.2. `temp_background_matrix = Foreground(frame)`;
 - 6.2.3. `Background_matrix = temp_background_matrix`;
 - 6.2.4. `Difference_matrix = Foreground_matrix - Background_matrix`;
 - 6.3. end if
 - 6.4. `if mod(track_refreshrate, i) ~= 0` /* background requires only a small update */
 - 6.4.1. `Foreground_matrix = Foreground (frame)`
 - 6.4.2. `temp_background_matrix = 0`;
 - 6.4.3. `Difference_matrix = Foreground_matrix - Background_matrix`;
 - 6.4.4. Perform Background modeling using Running Average method using updated value of Alpha returned by BBBCO procedure.
 - for `i=1:FrameHeight`
 - for `j=1:FrameWidth`
 - `if(Foreground_matrix(i,j) ~= 0)`
 - `Background_matrix (i,j) = (1-Alpha)* Background_matrix(i,j) + Alpha* Foreground_matrix(i,j)`;
 - end if
 - end for
 - end for
 - `Background(frame) = Background_matrix`;
 - 6.5. end if
 7. Generate **binary frame** on `Difference_matrix` using threshold values returned by BBBCO in step 5.
 8. **Apply Morphological Processing** for enhancing detection of objects
 - 8.1. Apply morphological open, close and fill operations to clear holes and enhance detections.
 - 8.2. Perform blob analysis to get objects of interest with bounding boxes and other parameters.
 9. **Plotting with bounding boxes:** Find contours and draw bounding boxes on detected objects.
 10. Set `i=i+1`; /* Loop at step 2 ends */
 11. end Loop
- The modified BBBCO-based BS method works by reading video sequences and extracting useful parameters such as frame rate, height, and width. The method initializes the background and foreground frames using the first frame. Moving objects are calculated thereafter by subtracting the current frame from the background frame. The proposed modified BS method calls the BBBCO procedure to find optimal of various parameters and threshold for the specific frame. The threshold value detected by BBBCO is used for pixel-wise comparison to separate foreground from background. The background refresh rate is modeled using another parameter whose value is governed by relative changes in the foreground and background regions and helps determine whether a background refresh is required. A learning parameter (Alpha) modeled by the BBBCO algorithm as the ratio between the optimal and maximum threshold values is also employed to update the background periodically. Once an object is detected, it is processed through a sequence of morphological operations to smooth the boundaries of the detected object and suppress holes and false detections. Blob analysis is performed on the detected objects to detect contours and bounding boxes, which are further used to plot objects on actual frames.
- The BBBCO algorithm generates an optimized threshold value for the modified BS algorithm using a total of nine steps (four major steps and five minor steps) discussed here. The first step, initialization, consists of generating a set of initial candidate solutions and the required variables are

distributed randomly with an upper bound. The pre-processing step includes all operations from reading a frame to finding the fuzzy entropies for the foreground and backward regions. The input frame is read and the probability distribution of each gray level is calculated and normalized in the range 0–1. Input frames are fuzzified into foreground and background regions using the initial population. The fuzzy 2-partition entropies of these regions are calculated using Shannon’s entropy, and the collective entropies of both regions are used to measure the fitness levels of potential candidates. The objective function is the overall entropy of both regions and the aim is to maximize its value. The maximization process iterates the BBBCO algorithm over several generations. Given a set of candidate solutions, the algorithm finds an effective center of mass for every iteration (step 3). In step 4, the next generation of candidate solutions is generated in accordance with the center of mass generated in the previous step. Additional randomness is incorporated into the population using a random function to generate new candidates. The center of mass generated in the previous step is also added to the population to prevent losing the best solution. This speeds convergence and avoids trapping the search in local minima. BBBCO goes through a series of iterations and stores the best candidate solution in variable S^{best} (step 6). Optimal values of parameters (a^* , b^* , c^*) are calculated from the best candidate, S^{best} (step 7). The value of these parameters is used to calculate the value of threshold using (14). Threshold along with other parameters, are returned by the algorithm in step 9. This threshold value is used to binarize the difference frame to extract moving object regions, and the other parameters are used to model the background.

Steps of BBBCO based parameter extraction procedure

1. Initialization: Initialize required variables.

- 1.1 Initialize BBBCO parameters:- Population size(N), number of iterations(M), number of parameters(p). /* for this algorithm p=3 for three variables a, b, c; N=30, M=40*/
- 1.2 Initialize population randomly in accordance to population size(N) and number of parameters (p) and store in variable [pop]_{Nxp} subject to constraints $0 \leq a \leq (L - 1)$; $0 \leq b \leq (L - 1)$; $0 \leq c \leq (L - 1)$ and $0 \leq a \leq b \leq c$.
- 1.3 Evaluate fitness for each candidate solution in variable [F]_{Nx1}
- 1.4 Each element in population is candidate for possible feasible solution and solution is represented as [S]_{Nx4} where each S_i represents some feasible values of a_i;b_i;c_i together with corresponding fitness value of solution.

Variables for population of candidates, fitness and solution space are represented as –

$$\text{pop}(N, p) = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \vdots & \vdots & \vdots \\ a_N & b_N & c_N \end{bmatrix}_{N \times 3} \quad \text{F}(N, 1) = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix}_{N \times 1}$$

$$S(N, 4) = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_N \end{bmatrix}_{N \times 4} = \begin{bmatrix} a_1 & b_1 & c_1 & F_1 \\ a_2 & b_2 & c_2 & F_2 \\ \vdots & \vdots & \vdots & \vdots \\ a_N & b_N & c_N & F_N \end{bmatrix}_{N \times 4}$$

2. Preprocess: Perform initial operations like generation of histogram and calculation of fuzzy entropies]

- 2.1 Read new frame as thisFrame and convert it to grayscale.
- 2.2 Normalize histogram of thisFrame between zero and one.
- 2.3: Find the number of pixels h(j) in thisFrame with gray level j, where $j = 0, 1, 2, \dots, (L-1)$
- 2.4: Find probability distribution of gray level values in frame using formula

$$p(j) = \frac{h(j)}{(\text{height} \times \text{width})}$$
- 2.5: Repeat For k=1: max number of iterations (M)
- 2.6: For each gray level value j, where $j = 1, 2, 3 \dots L - 1$, and for each value of a_i, b_i, c_i where $i = 1, 2, 3 \dots N$ from pop(i), perform fuzzification of foreground μ_{Fg}^k and background μ_{Bg}^k through parameters, using formulas (4) and (5).
- 2.7: Find pixel-wise fuzzy entropy for foreground and background regions defined as $H_{Fg}^k(i)$ and $H_{Bg}^k(i)$ using formulas (11) and (12).

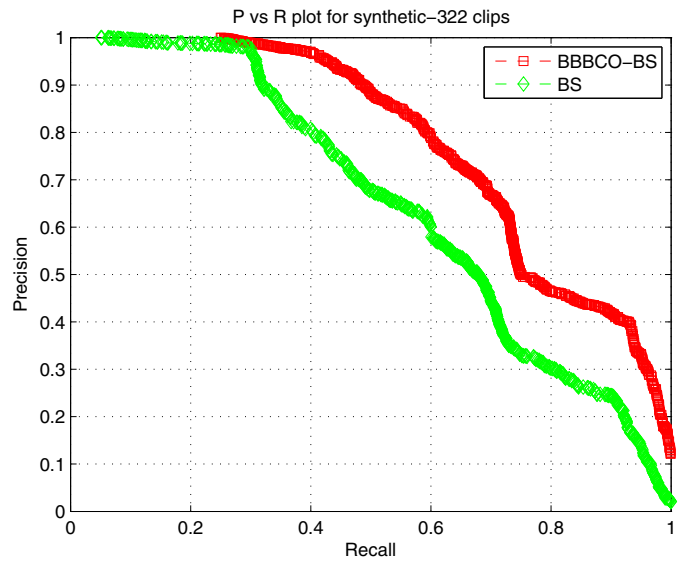
3. Fitness Evaluation: Calculate fitness of individuals and find center of mass.

- 3.1 Evaluate total fitness as sum of entropies of background and foreground

$$F^k(i) = H_{Fg}^k(i) + H_{Bg}^k(i)$$
- 3.2 Reorder all feasible solution candidates of the population in descending order, according to their respective fitness value $F^k(i)$
- 3.3 Calculate center of mass for current population of candidates at the kth iteration using formula

$$C^k(1) = \frac{\sum_{i=1}^N \frac{\text{pop}^k(i,1)}{F^k(i)}}{\sum_{i=1}^N \frac{1}{F^k(i)}}$$

Fig. 4 P vs R plot for synthetic-322 clips



where C_1^k represents the center of mass for k^{th} iteration and l refers to the value of one of the parameters and $pop^k(i, l)$ refers to i^{th} candidate solution.

4. Generation of new population of candidates

4.1 Generate new (N-1) set of candidate solutions computed randomly around center of mass using formula

$$pop(i, l)^{k+1} = C^k(l) \pm \frac{rand * (\max(pop^k(i, l)) - \min(pop^k(i, l)))}{k + 1}$$

rand is random number generator for generating random number between 0 and 1, max and min are used to select maximum and minimum values of respective variable from pool of candidates and $l = 1, 2, 3$ for variable a, b and c.

4.2 Copy best candidate of the previous step to new population's first index to prevent losing the best solution.

- 5. Set $k=k+1$ and return to Step 2.5.
- 6. Save the best solution as $S^{best} \leftarrow S(1, :)$.

Fig. 5 P vs R plot for wandering student video clip

- 7. Extract values of $(a^*b^*c^*)$ from the first three columns of S^{best} as $(a^*b^*c^*) = S(1, 1:3)$
- 8. Calculate final threshold from best solution obtained using formula

$$T = \begin{cases} a^* + \sqrt{\frac{(c^* - a^*)(b^* - a^*)}{2}} & (a^* + c^*) / 2 \leq b^* \leq c^* \\ c^* - \sqrt{\frac{(c^* - a^*)(c^* - b^*)}{2}} & a^* \leq b^* \leq (a^* + c^*) / 2 \end{cases}$$

- 9. Return threshold T with other parameters.
- 10. End

5 Performance evaluations

Evaluation of proposed algorithm is an important and complex task. Evaluation of such algorithms can be done by comparing the detected objects with ground truth values of the object. A large number of metrics have been proposed

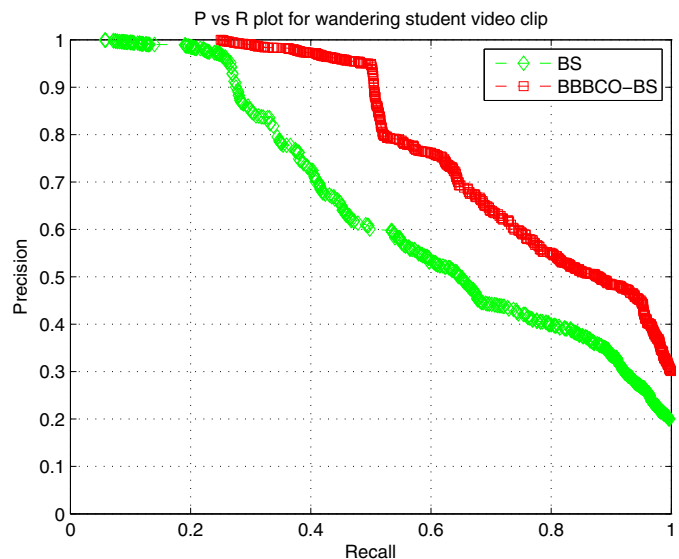
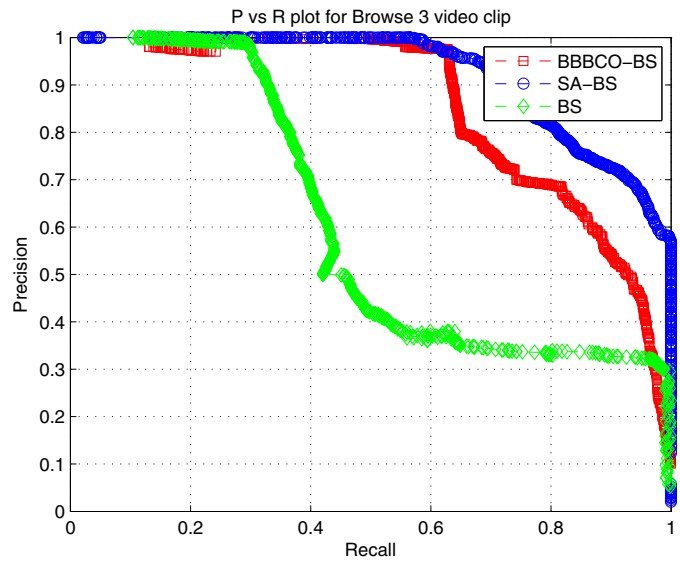


Fig. 6 P Vs. R plot for Browse3 video clip



in the literature and a few are discussed in this section for completeness. Nascimento and Marques [24] had proposed a framework for comparison of different motion detection algorithms based on false alarms, false failures, and correct detections. Readers may refer to studies of Karasulu et al. [25], Kasturi et al. [26] and McManus et al. [27] for detailed discussions on more metrics. These may be categorized as object and frame level metrics. Frame-based metrics are computed for every frame while object based metrics are based on coherency between ground truth object and detected object. Frame-based metrics include true positive rate sensitivity (TPR), false positive rate (FPR), false alarm ratio (FAR) and Specificity (Sp). Such metrics are computed from True positives (TP), False Positive (FP), False Negative (FN) and True Negatives (TN) as follows:

$$TPR = \frac{|TP|}{|TP| + |FN|} \tag{17}$$

$$FPR = \frac{|FP|}{|FP| + |TN|} \tag{18}$$

$$FAR = \frac{|FP|}{|TP| + |FP|} \tag{19}$$

$$Sp = \frac{|TN|}{|FP| + |TN|} \tag{20}$$

Where $|TP|$, $|FP|$, $|TN|$, $|FN|$ represents the number of objects identified as true positive, false positive, true negatives and false negatives respectively.

Apart from above defined metrics, five other frame based metrics have been used as Accuracy of Object Count (AOC), Precision (P), Recall (R), Average Fragmentation (AFR) and F-Measure (FM) (i.e., F-Score (FS)). Overall values for these metrics are obtained by averaging of all frame wise values. The range of these parameters goes from 0 to +1 with 1 being the best case. These are calculated as follows:

Fig. 7 P Vs. R plot for Browse4 video clip

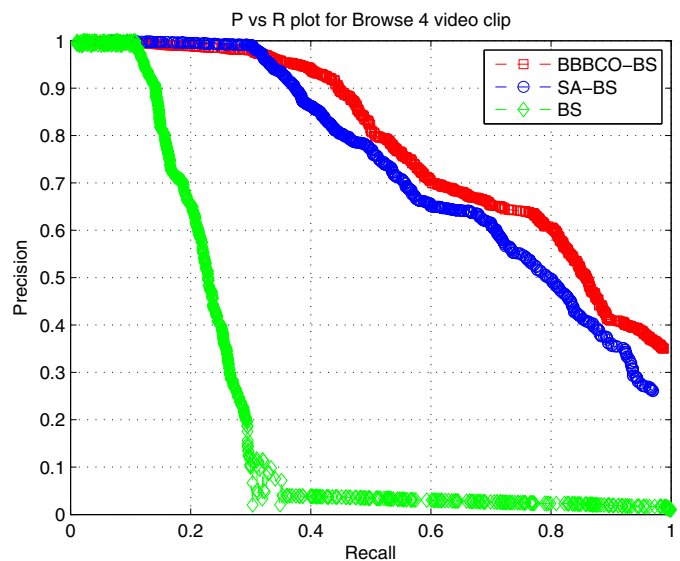
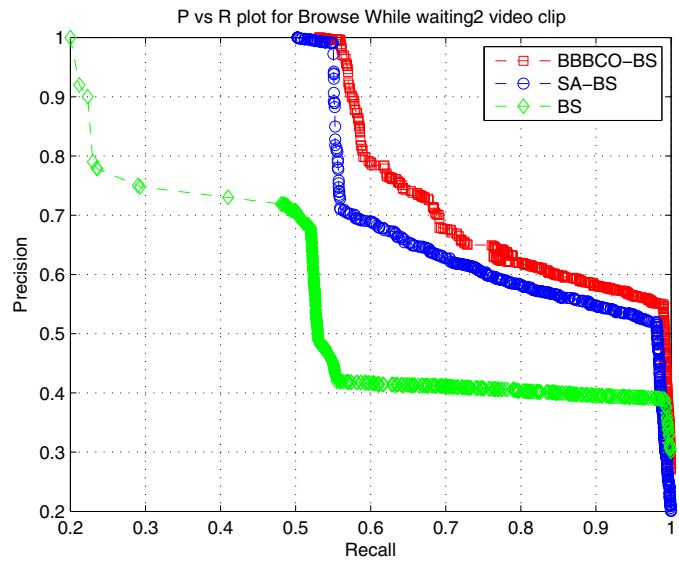


Fig. 8 P vs. R plot for Brw2 video clip



Let $GT(t)$ be the set of the GT objects in a single frame t and let $DT(t)$ be the set of output boxes, which are results of an algorithm in that frame. The AOC for frame t is defined as in (21)

$$AOC(t) = \frac{\text{Min}(|GT(t)|, |DT(t)|)}{|GT(t)| + |DT(t)|/2} \quad \text{if } |GT(t)| + |DT(t)| \neq 0 \quad (21)$$

where $|GT(t)|$ and $|DT(t)|$ are number of ground truth objects and number of boxes detected by proposed method at any frame t . In this measure, only the count of boxes in each frame is considered, but spatial information is not considered. When $|GT(t)| + |DT(t)| = 0$, it refers to the case when no object was present in the original frame and no object was detected by the algorithm as well. The value of AOC for that frame will simply be ignored; Precision for

any frame (t) is defined as the percentage of selected items that are correct as is calculated using formula given in (22):

$$\text{Precision}(t) = \frac{|TP(t)|}{|TP(t)| + |FP(t)|} \quad (22)$$

Recall is defined as the percentage of correct items that are selected and it is measured as

$$\text{Recall}(t) = \frac{|TP(t)|}{|TP(t)| + |FN(t)|} \quad (23)$$

The fragmentation of the output boxes overlapping a $GT(t)$ in frame t is defined as

$$\text{Frag}(GT(t)) = \frac{1}{1 + \log_{10}(|GT(t)| \cap |DT(t)|)} \quad \text{if } |GT(t)| \cap |DT(t)| \neq 0 \quad (24)$$

When $|GT(t)| \cap |DT(t)| = 0$; the value of fragmentation will be ignored as it refers to the case when no bounding

Fig. 9 P vs. R plot for FrA1 video clip

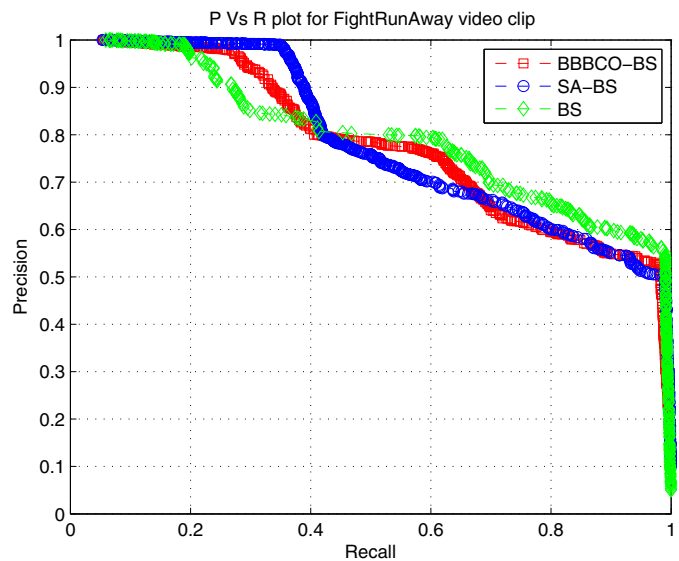


Table 1 Statistical comparison of various background subtraction methods

Video clip	Metrics											
	TPR			FPR			FAR			SP		
	BS	SA-BS	BBBCO-BS	BS	SA-BS	BBBCO-BS	BS	SA-BS	BBBCO-BS	BS	SA-BS	BBBCO-BS
Br3	0.745	0.868	0.879	0.367	0.412	0.361	0.23	0.34	0.19	0.633	0.588	0.639
Br4	0.437	0.779	0.776	0.378	0.313	0.267	0.45	0.28	0.26	0.622	0.687	0.733
BrW2	0.534	0.754	0.769	0.43	0.351	0.299	0.39	0.35	0.28	0.57	0.649	0.701
FrA1	0.789	0.735	0.765	0.201	0.256	0.231	0.27	0.32	0.35	0.799	0.744	0.769

boxes are present in the frame (t). Overall fragmentation value is calculated as the average of all frame level fragmentation values and is known as AFR (Average Fragmentation Rate). Often precision and recall measures are contradictory in nature and it is always advisable to evaluate any method based on a combination of these two measures. F-measure has been used for this purpose in a number of studies in the literature. It is also known as effectiveness measure and is defined as

$$FM = \frac{(\beta^2 + 1) \text{Precision} \cdot \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}} \tag{25}$$

In order to balance the weight of precision and recall, we substitute $\beta = 1$ that gives us balanced F-1 measure as

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})} \tag{26}$$

5.1 Experiment results

The current study has been tested on the benchmark BMC and Context Aware Vision using Image-based Active Recognition (CAVIAR) datasets, which are available at <http://bmc.iut-auvergne.com/> and <http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/>, respectively. The BMC dataset consists of 29 videos, categorized under two sets: learning and evaluation set. Videos under learning set are used for

tuning of parameters while videos under evaluation consist of more complex synthetic and real videos with truth values also bundled together. The CAVIAR dataset consists of video clips that are recorded via a wide-angle camera. CAVIAR dataset consist of 27 video clips captured in the INRIA lab entrance hall for testing purposes along with Ground truth(GT) values. A number of tests and evaluations have been carried out and these have been projected in this section. Plots for precision P(t) and Recall R(t) for all ‘t’ frames have been plotted in videos captured from the dataset and are shown in Figs. 4, 5, 6, 7, 8 and 9. Precision versus Recall feature is useful for evaluating the performance of a classifier. Ideal value for precision versus recall curve is on the upper right hand corner of plots which depicts 100 % precision and recall features of the classifier. Worst behavior of classifier is when precision and recall tend to zero and curves shift towards the bottom left corner. Statistical tests have also been conducted and are represented in the form of Tables 1 and 2.

Two random videos under evaluation set from the BMC dataset, namely “synthetic 322” and “wandering students” were chosen for testing. “Synthetic 322” consists of 1501 PNG files while “wandering students” consists of 795 frames together with ground truth values. Figures 4 and 5 show precision and recall, respectively, obtained for these sample videos. The results of proposed method are

Table 2 Performance comparison of various background subtraction methods

Video clip	Metrics														
	AOC			P			R			AFR			F		
	BS	SA-BS	BBBCO-BS	BS	SA-BS	BBBCO-BS	BS	SA-BS	BBBCO-BS	BS	SA-BS	BBBCO-BS	BS	SA-BS	BBBCO-BS
Br3	0.736	0.8484	0.8912	0.2431	0.6209	0.645	0.4512	0.8223	0.8511	0.6344	0.7223	0.8342	0.3164	0.7075	0.7843
Br4	0.3253	0.8615	0.87	0.6911	0.5456	0.617	0.3608	0.7142	0.7234	0.4208	0.7912	0.8548	0.474	0.6186	0.7012
BrW2	0.2919	0.2967	0.2861	0.473	0.3949	0.4452	0.6668	0.9602	0.9512	0.9841	0.9934	0.9879	0.5534	0.5596	0.5689
FrA1	0.9037	0.608	0.734	0.3658	0.5733	0.5612	0.4607	0.5129	0.6234	0.2023	0.3237	0.3674	0.4078	0.6444	0.3978

compared with the standard BS algorithm for both videos. For “wandering students,” in which a larger number of moving objects are present in a single frame, BBBCO-based BS obtains good results. Relatively higher values of precision and recall are observed in the graphs because the modified BS algorithm could extract finer details of detected objects and model the background effectively. The BBBCO-based BS algorithm could make a large number of detections with correct object boundaries, thereby moving the results towards the upper right corner of the graph.

The proposed study was also evaluated on videos from the CAVIAR dataset and compared with a similar method by Karasulu [19] and the standard BS algorithm. The same video clips used in [19], namely “Browse3,” “Browse4,” “Browse_WhileWaiting2,” and “Fight_Run Away1” are used in this evaluation. These are referred to as Br3, Br4, BrW2, and FrA1 in the present study and consist of 912, 1,140, 1,896 and 552 video frames, respectively. Figures 6–9 show plots of Precision versus Recall values for the four video sequences chosen from the CAVIAR dataset. The curves of the SA-BS and BBBCO-BS method are similar for almost all the videos. However, the proposed BBBCO-based BS method performs better than the simple BS algorithm and the SA-based BS method because the values of precision and recall in the case of the proposed approach are higher. The total number of misses by the standard BS algorithm is higher while almost the same number of detections are obtained by both entropy-based methods. For Br3 and Br4, the proposed method outperforms both SA-BS and simple BS, and is able to make detections that are more accurate (Figs. 6 and 7). For the Brw2 video clip, the number of detections obtained by the proposed method are significantly higher than those obtained by both comparison methods, as is clear in Fig. 8. This can be attributed to the fact that the proposed scheme could detect slow-moving objects and those that with a small size efficiently. For the FrA1 video clip, the detection of all the methods was almost same, as there were fewer intensity variations in the video, and even the simple BS method was competitive with respect to other methods and obtained good results. Overall, it can be concluded that the proposed approach is still preferable to other methods for effective detection.

Various statistical measures were also calculated for the proposed method and the corresponding results are shown in Tables 1 and 2. Table 1 lists the values of the true positive rate (TPR), false positive rate (FPR), false alarm rate (FAR) and specificity (Sp). The values of AOC, P, R AFR, and F are shown in Table 2. Larger values of these variables indicate a better algorithm. Both tables clearly show that the proposed method is preferable to the rest of the methods. The proposed methodology can simulate the effects of changing illumination, dynamic backgrounds, and noisy environments. The number and accuracy of the detections

obtained by the proposed method are higher for almost all video clips.

6 Conclusion

This study presented a novel method for object detection in videos using the fuzzy entropy principle and BBBCO. The study proposed a new variant of the standard BS algorithm to extract objects from video sequences by employing the concept of fuzzy partition entropy. The modified algorithm determines the optimal threshold value for classifying pixels as foreground and background. This is done using the concept of fuzzy 2-partition entropy and framing the problem of threshold detection as an optimization problem. A recent optimization technique called BBBCO is used to detect the effective threshold. The proposed technique is novel, as the concept of BS has not been explored using fuzzy entropy and optimization theory. The proposed technique was tested on benchmark datasets and a number of metrics were evaluated to gauge its effectiveness. The method was also compared with the traditional BS method and another recent method. The obtained results are highly encouraging and show improvement over state-of-the-art methods.

References

1. Ling Q, Yan T, Li F, Zhang Y (2014) A background modeling and foreground segmentation approach based on the feedback of moving objects in traffic surveillance systems. *Neurocomputing* 133(8):32–45
2. Tian Y, Senior A, Max L (2012) Robust and Efficient Foreground Analysis in Complex Surveillance Videos. *Mach Vis Appl* 23(5):967–98
3. Arroyo R, Yebes J, Bergasa LM, Daza IG, Almazán J (2015) Expert Video-Surveillance System for Real-Time Detection of Suspicious Behaviors in Shopping Malls. *Expert Syst Appl* 42(21):7991–8005
4. Makris D, Ellis T (2002) Path detection in video surveillance. *Image Vis Comput* 20(12):895–903
5. Shakeri M, Zhang H (2012) Real-Time Bird Detection Based on Background Subtraction. In: *Proceeding of World Congress on Intelligent Control and Automation at Beijing, China*, pp 4507–4510
6. Heikkila J, Silven O (1999) A Real-Time System for Monitoring of Cyclists and Pedestrians. In: *2nd IEEE Workshop on Visual Surveillance at Fort Collins, USA*, pp 74–81
7. Mandellos NA, Keramitsoglou I, Kiranoudis CT (2011) A background subtraction algorithm for detecting and tracking vehicles. *Expert Syst Appl* 38(3):1619–1631
8. Yoshinaga S, Shimada A, Nagahara H, Taniguchi R-I (2014) Object detection based on spatiotemporal background models. *Comput Vis Image Underst* 122:84–91
9. Chen Z, Ellis T (2014) A self-adaptive Gaussian mixture model. *Comput Vis Image Underst* 122:35–46
10. Spampinato C, Palazzo S, Kavasidis I (2014) A texton-based kernel density estimation approach for background modeling under extreme conditions. *Comput Vis Image Underst* 122:74–83

11. Bouwmans T (2014) Traditional and recent approaches in background modeling for foreground detection: an overview, *Computer Science Review* 11-12, 31–66
12. Shaikh SH, Saeed K, Chaki N (2014) Moving Object Detection Using Background Subtraction. In: Springer Briefs in Computer Science, Springer International Publishing, pp 1–6. ISBN: 978-3-319-07385-9
13. Piccardi M (2004) Background subtraction techniques: a review. In: IEEE international conference on systems, man and cybernetics at the Hague, The Netherlands 4, pp 3099–3104
14. Sobral A, Vacavant A (2014) A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Comput Vis Image Underst* 122(6):4–21
15. Karasulu B, Korukoglu S (2013) Moving object detection and tracking in videos: Performance evaluation software. Springer briefs in computer science, 2013, Springer-Verlag New York, pp. 1–30, ISBN: 978-1-4614-6533-1
16. Li-juan Q, Yue-ting Z, Fei W, Yun-he P (2005) Video segmentation using Maximum Entropy Model. *J Zhejiang Univ (Sci)* 6(1):47–52
17. Wang F-P, Chungy W-H, Kuo S-Y (2012) An efficient approach to extract moving objects by the h.264 compressed-domain features. In: 12th International Conference on ITS Telecommunications at Taipei, pp 452–456
18. Subudhi BN, Nanda PK, Ghosh A (2011) Entropy based region selection for moving object detection. *Pattern Recogn Lett* 32:2097–2108
19. Ma Y-F, Zhang H-J (2001) Detecting Motion object by spatio-temporal Entropy. In: IEEE International Conference on Multimedia and Expo, pp 265–268
20. Karasulu B, Korukoglu S (2012) Moving object detection and tracking by using annealed background subtraction method in videos: Performance optimization. *Expert Syst Appl* 39:33–43
21. Erol OK, Eksin I (2006) A new optimization method: Big Bang-Big Crunch. *Adv Eng Softw* 37:106–111
22. <http://web.itu.edu.tr/okerol/BBBC.html> last accessed on April 2016
23. Tang H, Zhou J, Xue S, Xie L (2010) Big bang–big crunch optimization for parameter estimation in structural systems. *Mech Syst Signal Process* 24(8):2888–2897
24. Nascimento JC, Marques JS (2006) Performance evaluation of object detection algorithms for video surveillance. *IEEE Trans Multimedia* 8(4):761–774
25. Karasulu B, Korukoglu S (2010) A software for performance evaluation and comparison of people detection and tracking methods in video processing. *Multimed Tool Appl* 11:205–218
26. Kasturi R, Goldgof D, Soundararajan P, Manohar V, Garofolo J, Bowers R et al (2009) Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol. *IEEE Trans Pattern Anal Mach Intell* 31(2):319–336
27. Lazarevic-McManus N, Renno JR, Makris D, Jones GA (2008) An object-based comparative methodology for motion detection based on the F-measure. *Special Issue on Intell Visual Surveillance Understanding* 111(1):74–85
28. Cheng HD, Chen JR, Li JG (1998) Threshold selection based on fuzzy c-partition entropy approach. *Pattern Recogn* 31: 857–870
29. Tang Y, Mu W, Zhang Y, Zhang X (2012) A fast recursive algorithm based on fuzzy 2-partition entropy approach for threshold selection. *Neurocomputing* 74:3072–3078

Manisha Kaushal received her Bachelors degree in Computer Science & Engineering from Punjab Technical University, Jalandhar, Punjab, India in 2006 and Masters degree in Computer Science and Engineering in 2010 from Punjab Technical University, Jalandhar, Punjab, India. She is pursuing Ph.D. in Computer Science and Engineering under the guidance in the area of soft computing and computer vision. She is currently working as an Assistant Professor in Computer Science and Engineering at University Institute of Engineering and Technology, Panjab University, Chandigarh, India. Her research interests include computer vision, video analysis and soft computing techniques.

Baljit Singh Khehra received his Bachelor of Engineering degree in Computer Engineering from Punjab Technical University, Jalandhar, Punjab, India in 1998, the Master of Technology in Computer Science and Engineering from Punjabi university, Patiala and the Doctor of Philosophy from Sant Longowal Institute of Engineering and Technology Deemed University (established by Govt. of India), Longowal, Sangrur, Punjab, India in 2014. Currently, he is an Associate Professor in the Department of Computer Science and Engineering of Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib, Punjab, India. He has published more than 65 research papers in various international journals and refereed conferences. His main research interests include image processing, medical imaging, artificial neural networks and genetic algorithms. He is a fellow of IE (I), artificial neural networks and genetic algorithms. He is a fellow of IE (I), life member of CSI, ISTE, IETE and IAENG.