Motion detection using block based bi-directional optical flow method

Sandeep Singh Sengar, Susanta Mukhopadhyay

PII: S1047-3203(17)30170-0
DOI: http://dx.doi.org/10.1016/j.jvcir.2017.08.007
Reference: YJVCI 2045

To appear in: *J. Vis. Commun. Image R.*

Received Date: 24 November 2016
Accepted Date: 17 August 2017

Please cite this article as: S.S. Sengar, S. Mukhopadhyay, Motion detection using block based bi-directional optical flow method, *J. Vis. Commun. Image R.* (2017), doi: http://dx.doi.org/10.1016/j.jvcir.2017.08.007

Motion detection using block based bi-directional optical flow method

Sandeep Singh Sengar*, Susanta Mukhopadhyay

Department of Computer Science and Engineering
Indian Institute of Technology (Indian School of Mines),
Dhanbad, India-826004

Abstract

Detecting moving objects from video frame sequences has a lot of useful applications in computer vision. This proposed method of moving object detection first estimates the bi-directional optical flow fields between (i) the current frame and the previous frame and between (ii) the current frame and the next frame. The bi-directional optical flow field is then subjected to normalization and enhancement. Each normalized and enhanced optical flow field is then divided into non-overlapping blocks. The moving objects are finally detected in the form of binary blobs by examining the histogram based thresholded values of such optical flow field of each block as well as the optical flow field of the candidate flow value. Our technique has been conceptualized, implemented and tested on real video data sets with complex background environment. The experimental results and quantitative evaluation establish that our technique achieves effective and efficient results than other existing methods.

Keywords: Optical flow, motion detection, normalization, block, morphology

1. Introduction

Motion detection is extensively used in computer vision to facilitate the analysis of real-world video scenes. Particularly, some typical computer vision

*Corresponding author: Tel.: +91 8804923594
Email address: sandeep.iitdhanbad@gmail.com (Sandeep Singh Sengar)

applications deal with monitoring traffic and pedestrian [1, 2], automatic robot navigation [3], target counting [4], detection of human activity [5, 6]. The aim of moving object detection is to discriminate the pixels of moving foreground objects from the background.

Moving object detection techniques [7] can be divided into three categories: (i) frame differencing [8] (ii) background subtraction [9] and (iii) optical flow [10]. The moving objects are detected using the frame differencing approach by computing the pixel-wise difference between the consecutive frames of video sequences. This technique can efficiently detect the objects in different challenging environments, but sometimes creates holes inside the foreground due to slow movement or pause of the foreground objects, ghosting and foreground aperture problems. Background subtraction approaches also efficiently detect the moving objects with the steps of background modeling and thresholding function. However, this method fails in different scenarios due to lack of generated background model. Next, the flow vector based optical flow approach is used to compute the apparent motion of object between consecutive frames. This method is highly robust in detecting the moving target regions, but, it gives inaccurate results with the slow motion objects and illumination variation environments. Therefore, we propose a robust motion detection technique with the help of optical flow to efficiently and accurately detect the foreground objects in different challenging environments.

The rest of this paper is organized as follows. After this introductory section the related work is provided in Section 2. Motivation and contributions are given in Section 3. The proposed block based normalized optical flow method is elaborated in Section 4. Experimental results and analysis are provided in Section 5. Finally, Section 6 concludes our work.

2. Related work

Recently, lots of techniques for motion detection have been proposed by computer vision groups [11, 12]. Here in the following sections, we will focus our overview of related works with the frame differencing, background subtraction and optical flow techniques.

2.1. Frame differencing

A method for detecting moving people in the indoor environments is proposed with the help of frame differencing and neural network based classification techniques [13]. This method reduces the false alarm and provides a robust classification with the help of a finite state automation. A pedestrian detection method using region of interest (ROI) and multi-block local binary pattern descriptors is proposed by Halidou et al. [14]. Here ROI is estimated with the help of optical flow and three frame differencing method. A human detection technique based on optical flow and frame differencing for a thermal infrared camera mounted on an autonomous mobile robot is proposed by Caballero et al. [15]. The frame differencing and the non-pyramidal Lucas-Kanade approaches [16] are used to detect human candidates based on thermal signatures when the robot stops and moves. A method is proposed with the help of modified temporal differencing and local fuzzy thresholding techniques to robustly detect the moving objects in the presence of different challenging environment like dynamic backgrounds, shadows and bad weather [17]. To detect the motion in the moving camera environment, a temporal differencing based integration scheme is proposed with the help of feature points and foreground regions [18].

2.2. Background subtraction

Several foreground detection techniques based on the background subtraction and modeling are used in the literature [19, 20]. A self-organizing technique for background subtraction is proposed by Maddalena et al. [21]. In this technique, neural network (and labeling decision) at neighboring pixels are used to learn the background (and update background). A background modeling approach is proposed by projecting the high-dimensional data into a lower dimensional subspace with the help of PCA [22]. This method is appropriate for global illumination changes, but it does not accurately detect the slow motion objects with local illumination variations. An approach based on weighted mixture of Gaussians is provided to model the distribution of the values observed over time for each pixel [23]. However, this approach is not appropriate for rapidly changing environment, like sudden illumination changes and non-static backgrounds. A real time visual surveillance system W4 is used to detect and track the objects from the monocular gray-scale video captured by an infrared camera [24]. A real time hybrid background modeling based vehicle detection technique is proposed by utilizing the motion information of different activities in the scene [25]. The motion vector based moving object detection method is used to find out the blocks that contain motion [26]. This method has employed the frame differencing, background subtraction and optical flow techniques to determine the moving objects in the High Efficiency Video Coding (HEVC) compressed video [27, 28].

2.3. Optical flow

A method based on optical flow [10] gives a way to represent and determine motion within a frame of the video. A total variation technique for the robust estimation of the global optical flow vector field is presented by Horn and Schunck [29]. A modified optical flow method is proposed using quaternion to decrease the inaccuracy due to unclear pixel information [30]. To estimate the position of the target object, Schwarz et al. [31] utilized depth information acquired by a time of flight camera. Both the above methods [30, 31] provide accurate results and reduce processing time. However, they do not produce good results for real time applications. A visual object tracking approach using adaptive neuro fuzzy inference system for arbitration between finite impulse response and optical flow is proposed by Choi et al. [32]. A method is proposed for motion classification and clustering of the video frames captured by a freely moving camera [33]. This method removes scattered points (outliers) in the flow field by combining the random sample consensus (RANSAC) and optical flow technique. The flow field is transformed into polar co-ordinates (magnitude and orientation) from the cartesian coordinates, before it is divided into blocks. The initial number of clusters and the corresponding cluster centers are obtained by counting the selected blocks and estimating the density of the moving points respectively. An optical flow estimation method based on genetic model is proposed by Tagliasacchi [34]. This method employs the watershed technique to segment each frame of the video data based on similarity in color and spatial proximity. This method outperforms the Lucas-Kanade [16] at the object boundaries. However, the processing time of this method is high and it takes more than 1 second to segment a 176×144 frame using a Pentium M 1.6 Ghz, which is not suitable for real time applications. A method for moving object area detection using normalized self adaptive optical flow is proposed by Sengar et al. [35]. This approach adapts the threshold value with the help of Otsu's [36, 37] method and selects the moving object area using self adaptive window approach. It is an improvement over [38] for moving object detection. However, this approach does not perform well with moving objects of small size.

3. Motivation and contributions

Several techniques have been proposed over the time to find accurately the moving objects from the image sequences. However some of the major challenges in such problems is detection of slowly moving objects of small size, illumination variation and noisy environment. It is a typical task to accurately detect the small and slowly moving objects due to their small motion area. Some of the existing algorithms cannot be implemented for real time due to its high computational cost. To address these issues, we have attempted to develop a block based bi-directional optical flow (BBOF) technique to effectively and efficiently detect such moving objects. In this work, first we employed Gaussian filter to suppress the noise from video frames, then estimated the optical flow across (i) the previous frame and the current frame and (ii) the current frame and the next frame separately. We combine both the optical flow components, apply (a) normalization (b) block based binarization (c) morphological operation, and (d) connected component analysis to complete the task of detecting the moving objects. We have tested our algorithm on different publicly available real video datasets that produce effective and efficient results with relatively low computational overhead.

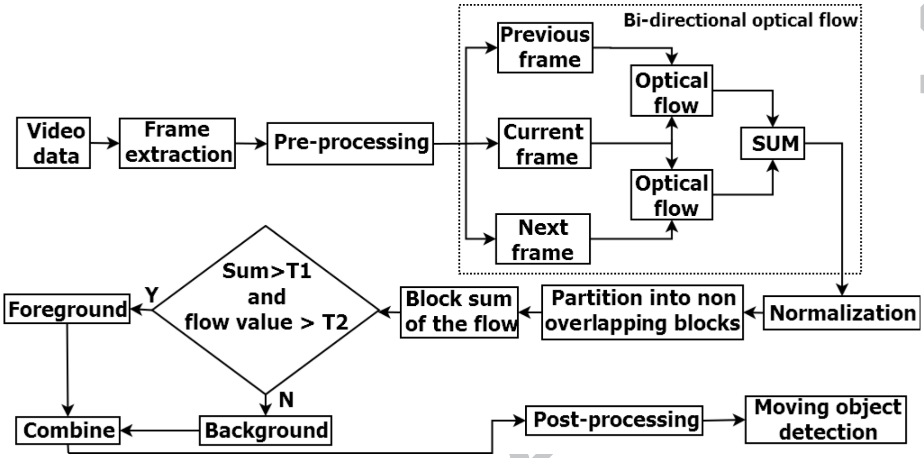


Figure 1: Schematic diagram of the proposed approach

4. Proposed method

The proposed method of motion detection (i) extracts the frames from the video data (ii) converts it into gray-scale, if it is in color format (iii) suppresses the noise by applying Gaussian filter on individual frame, before it estimates the optical flow independently between each pair of (a) the current frame and its previous frame (OF_1) and (b) the current frame and its next frame (OF_2). The bi-directional optical flow (OF) of each video frame is then computed by algebraically summing up the results of the optical flow components OF_1 and OF_2. Subsequently, normalization step is employed to enhance the information content in the bi-directional optical flow. The resultant optical flow in the next step is divided into a number of non-overlapping blocks. The moving object is then detected by examining each block with respect to (a) the individual flow value of the candidate pixel as well as (b) the sum of flow values computed over the block. A post processing step consisting of a morphological operation and connected component analysis are executed so as to eliminate the effect of spurious motion components. The schematic diagram of the proposed technique is given in Fig. 1. Below we elaborate the sequence of steps for the proposed method.

4.1. Pre-processing

In the first step of preprocessing, we extract and keep as individual frames from the three dimensional input video data *D(x,y,t)*, where the i^{th} frame is $F_i(x, y)$= $D(x, y, t = t_i)$. For the time being we focus on gray-scale frames instead of the color ones as because: a gray-scale frame is much easier to process as compared to the three dimensional structure of color frames. So in the next step, we convert the color video frames to gray-scale using:

$$F_i(x, y) = 0.2126R_i(x, y) + 0.7152G_i(x, y) + 0.0722B_i(x, y) \qquad (1)$$

Here $R_i(x, y)$, $G_i(x, y)$ and $B_i(x, y)$ represent the red, green and blue components of each pixel for i^{th} color frame respectively, and scalar multiples are the corresponding weights of these components.

Subsequently, we employ smoothing operation on the individual frame for suppressing noise. We have applied a cascaded combination of two 1-D Gaussian filters (G) [39] along the horizontal and vertical direction respectively. Gaussian filter (G) can be employed to a video frame (F_i) as given below:

$$F_i^{smooth} = \frac{1}{\sigma\sqrt{2\pi}} \int \left[\int F_i(x, y) e^{-\left(x^2+y^2\right)/2\sigma^2} dx \right] dy \qquad (2)$$

4.2. Optical flow

Apparent motion of objects in the image sequences can be represented by optical flow. It can be approximated on frame sequences having small displacements and smooth motion fields. The basic idea of optical flow is to display the displacement of patterns in the video frames as a vector field, named as an optical flow field. Suppose $F(x, y, t)$ and $F(x + \Delta x, y + \Delta y, t + \Delta t)$ are the intensities of a pixel in two consecutive frames obtained at a time interval Δt

with positions (x, y, t) and $(x+dx, y+dy, t+dt)$ respectively. According to the brightness constancy assumption, we have

$$F_{t=i}(x, y) = F_{t+\Delta t}(x + \Delta x, y + \Delta y) \tag{3}$$

Using Taylor series expansions [40] at the right hand side of Eq. 3 and neglecting the higher order terms, we get

$$F_{t+\Delta t}(x + \Delta x, y + \Delta y) = F_{t=i}(x, y) + F_i^x \Delta x + F_i^y \Delta y + F_i^t \Delta t \tag{4}$$

Here F_i^x, F_i^y, and F_i^t denote the derivatives of F_i with respect to x, y, and t. Eq. 4 can be rewritten as (using Eq. 3):

$$F_i^x \Delta x + F_i^y \Delta y + F_i^t \Delta t = 0 \tag{5}$$

$$F_i^x u_i + F_i^y v_i + F_i^t = 0 \tag{6}$$

Here $u_i = \Delta x/\Delta t, v_i = \Delta y/\Delta t$ are the velocity components in horizontal and vertical direction respectively. Eq. 6 is the indeterminate equation of OF constrained with two unknowns u_i and v_i. Hence, the OF velocity field $(u_i,\ v_i)$ cannot be computed locally with the help of one equation, so additional constraints are required to solve it. To keep this problem in mind, Horn-Schunk [29] provided one more constraint based on smoothness to find the unique solution of the optical flow components. This constraint can be expressed by minimizing the quadratic sum of the gradient of OF velocity components, i.e.

$$min \left\{ \left(\frac{\partial u_i}{\partial x}\right)^2 + \left(\frac{\partial u_i}{\partial y}\right)^2 + \left(\frac{\partial v_i}{\partial x}\right)^2 + \left(\frac{\partial v_i}{\partial y}\right)^2 \right\} \tag{7}$$

Horn and Schunck [29] combined and minimized both the brightness and smoothness constraints to estimate the optical flow field:

$$H_i = \int\int (F_i^x u_i + F_i^y v_i + F_i^t)^2 + \alpha^2 \left\{ \left(\frac{\partial u_i}{\partial x}\right)^2 + \left(\frac{\partial u_i}{\partial y}\right)^2 + \left(\frac{\partial v_i}{\partial x}\right)^2 + \left(\frac{\partial v_i}{\partial y}\right)^2 \right\} dxdy \tag{8}$$

Here signal to noise ratio based regularizing parameter α^2 is used to weigh the smoothness in OF velocity field. We minimize Eq. 8 using the Gauss-Seidel iterative solution and obtain the following velocity components:

$$u_i = u_i^{av} - F_i^x \frac{Q_i}{L_i} \tag{9}$$

$$v_i = v_i^{av} - F_i^y \frac{Q_i}{L_i} \tag{10}$$

where $Q_i = F_i^x u_i^{av} + F_i^y v_i^{av} + F_i^t$, and $L_i = \alpha^2 + (F_i^x)^2 + (F_i^y)^2$
Here u_i^{av} and v_i^{av} represent the neighborhood mean of each pixel of u_i and v_i respectively.

We take three consecutive frames namely (1) the current $F_i(x,y) = F_i^{smooth}(x,y)$ (2) the previous $F_{i-1}(x,y)$=$F_{i-1}^{smooth}(x,y)$ and (3) the next $F_{i+1}(x,y)$=$F_{i+1}^{smooth}(x,y)$. Successively we estimate (1) the optical flow ($u_i^1(x,y)$, $v_i^1(x,y)$) across the frames $F_{i-1}(x,y)$ and $F_i(x,y)$ as well as (2) the optical flow ($u_i^2(x,y)$, $v_i^2(x,y)$) across the frames $F_i(x,y)$ and $F_{i+1}(x,y)$ independently using Eq. 9 and Eq. 10. Now, we compute the bi-directional optical flow by combining both the optical flow components:

$$U_i(x,y) = u_i^1(x,y) + u_i^2(x,y) \tag{11}$$

$$V_i(x,y) = v_i^1(x,y) + v_i^2(x,y) \tag{12}$$

4.3. Normalization

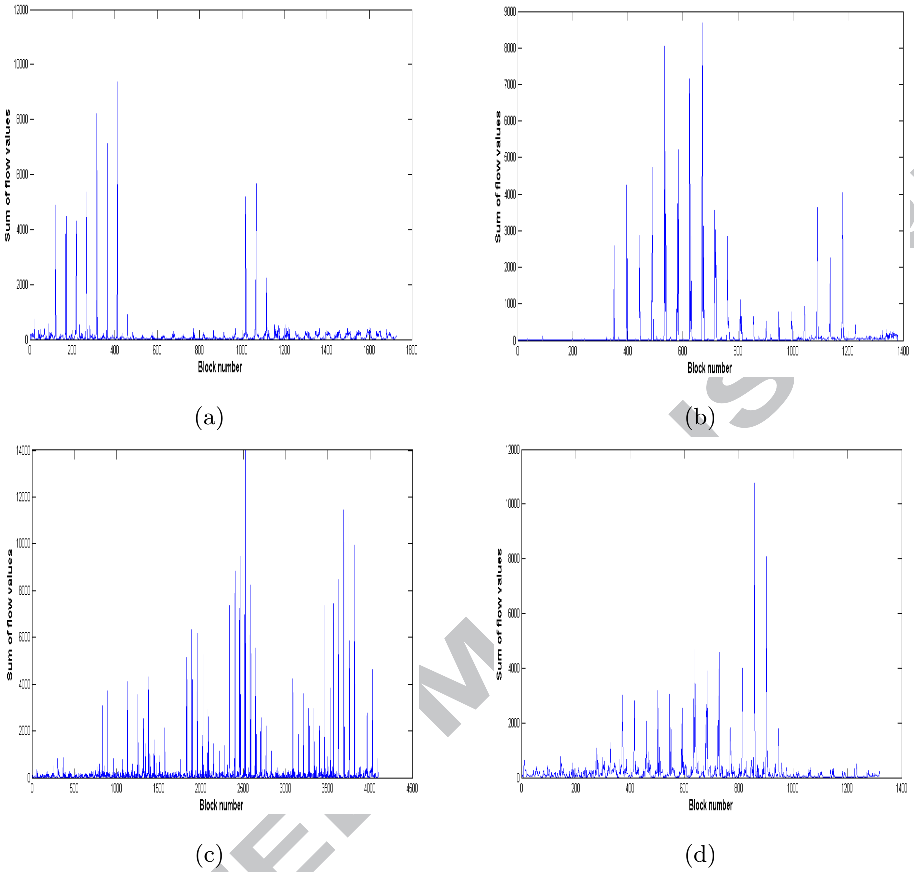
The magnitude of optical flow for the i^{th} frame can now be expressed as:

$$Fl_i(x,y) = \sqrt{U_i(x,y)^2 + V_i(x,y)^2} \tag{13}$$

The optical flow presented by Eq. 13 is shown in the form of 2-D gray-scale video frames. However, the optical flow's intensity values do not cover the range 0 to *L-1* (255 for 8-bit gray scale images) exhaustively. For this we can employ a linear stretching technique as given by Eq. 14 to stretch the intensity range of optical flow corresponding to each frame.

$$F_i^{ln}(x,y) = \left[\frac{Fl_i(x,y) - Fl_i^{min}}{Fl_i^{max} - Fl_i^{min}} \times 255 \right] \tag{14}$$

Here Fl_i^{min} and Fl_i^{max} are the minimum and maximum intensity values of magnitude of optical flow Fl_i.



(a)

(b)

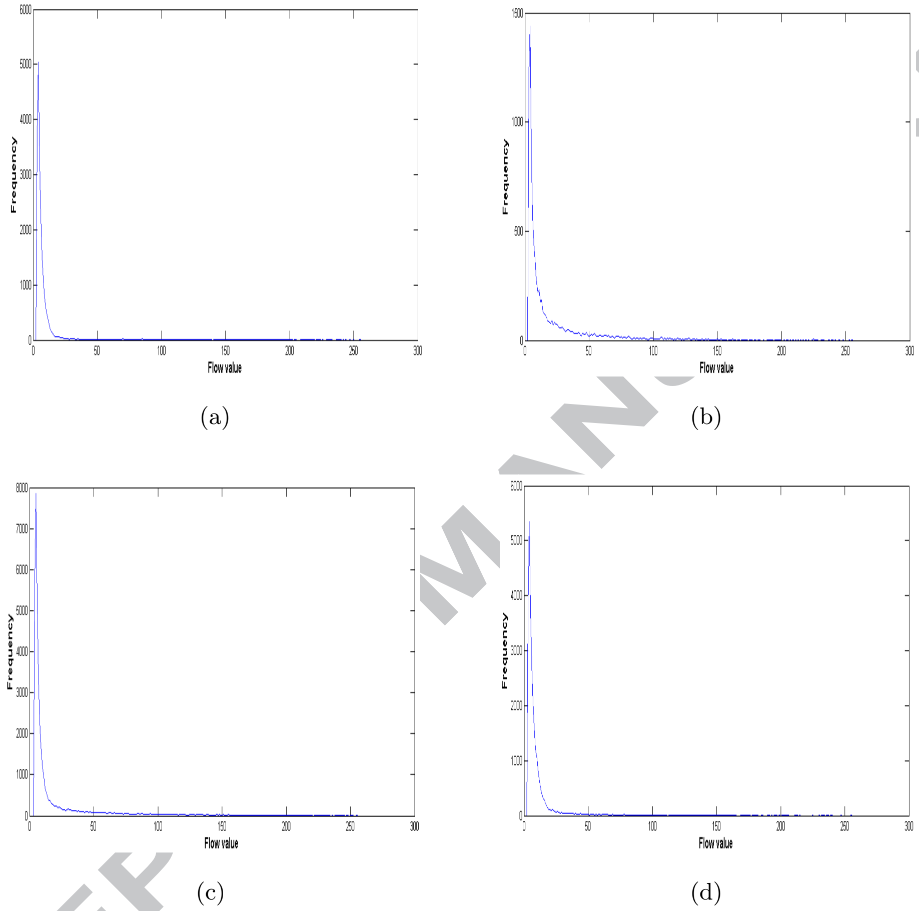
(c)

(d)

Figure 2: The plot of block sum of representative frames: (a) frame number 264 of video 1; (b) frame number 45 of video 2; (c) frame number 4 of video 3 (d) frame number 72 of video 4 .

4.4. Block processing on normalized optical flow field and binarization

Now we divide the normalized optical flow field into non-overlapping blocks B_{ij}^{ln}, subsequently, we estimate the block sum by adding the flow field at each pixel location in the block. Next, we classify the values of block as foreground or background by comparing the block sum and block's value against two prescribed threshold T_1 and T_2 respectively. The value of T_1 is selected by plotting block sum for each block against the block number, similarly T_2 is selected by the histogram of flow values. In Fig. 2 and Fig. 3, we have shown the corresponding plot of block sum and histogram of flow values respectively of representative frames of each video data we have experimented with.



(a)

(b)

(c)

(d)

Figure 3: The histogram of flow field of a representative frame: (a) frame number 264 of video 1; (b) frame number 45 of video 2; (c) frame number 4 of video 3 (d) frame number 72 of video 4 .

If we select high value of T_1 then most of the foreground pixel blocks will be lost due to the few number of peaks, conversely if we select a low threshold value then it will also detect some of the background pixel blocks as foreground. For the threshold value T_2, we are inspecting the histogram of flow values as given in Fig. 3. It is observed that for the particular frame the number of background pixels is much higher than that of the foreground pixels and the flow value corresponding to the background pixels are low compared to that of the foreground pixels. Thus the histogram shows a prominent peak at low flow values. The threshold T_2 is fixed in such a way that we can avoid the flow peaks corresponding to the background. If both the block sum and block value are more than the corresponding threshold then we set that pixel value to 1 otherwise we set it to 0.

$$FG_i(x,y) == \begin{cases} 1, & sum(B_{ij}^{ln}) > T_1 \text{ and } F_i^{ln}(x,y) > T_2 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Here FG_i, $sum(B_{ij}^{ln})$ represent the moving objects of the i^{th} frame, sum of the flow values of the j^{th} block of the i^{th} frame respectively. The flow value of i^{th} frame F_i^{ln} is estimated in Eq. 14.

4.5. Post-processing

We suppress the noise and finally detect the moving objects from the binary frame FG_i (obtained by Eq. 15) using morphological operation and connected component analysis. First to remove narrow constrictions, morphological opening operation (as shown in Eq. 16) is applied with square structuring element SE of size 5×5. In mathematical morphology, opening is cascaded combination of first erosion followed by dilation [41].

$$MV_i = (FG_i \ominus SE) \oplus SE \tag{16}$$

Each individual blob of the binary image is tagged with different labeled values. Thus the labels of two blobs which are not connected are also different. The number of pixels having the same tag label gives the area or size of the blob. If the blob size is too small, we can eliminate that blob by thresholding operation. Thus isolated noisy blobs are eliminated from the final image to obtain the accurate foreground objects.

5. Experimental results and analysis

To evaluate the effectiveness, our BBOF method has been tested on four standard benchmark real video datasets namely Video 1 (meeting) [42], Video 2 (walking) [43], Video 3 (traffic) [44], and Video 4 (hall monitoring) [45]. The detailed description of these data sets is as follows.

- *Video 1 [42]:* In this color video, two persons come from opposite sides and shake their hands and move together. This video contains small sized objects with bright and dark regions in the background, and also has varied illumination. There are 716 total number of frames in this dataset with size 288 pixels × 384 pixels (i.e. row size × column size). The used frame rate and bit rate of this dataset is 25 frames/sec and 24 kbps respectively.

- *Video 2 [43]:* In this color video, a person is moving slowly. The foreground is brighter than the background. There are total number of 132 frames with size 240 pixels × 368 pixels (i.e. row size × column size) are in this dataset. The frame rate and bit rate of this video is 25 frames/sec and 24 kbps respectively.

- *Video 3 [44]:* In this gray scale outdoor video data, a person is moving slowly on the footpath. The foreground contains a varying number of fast moving cars. The background contains trees and other natural objects in the form of dark and bright regions with small variation in intensity. There are 49 total number of frames in this video sequence with size 512 pixels × 512 pixels (i.e. row size × column size). The used frame rate and bit rate of this dataset is 30 frames/sec and 24 kbps respectively.

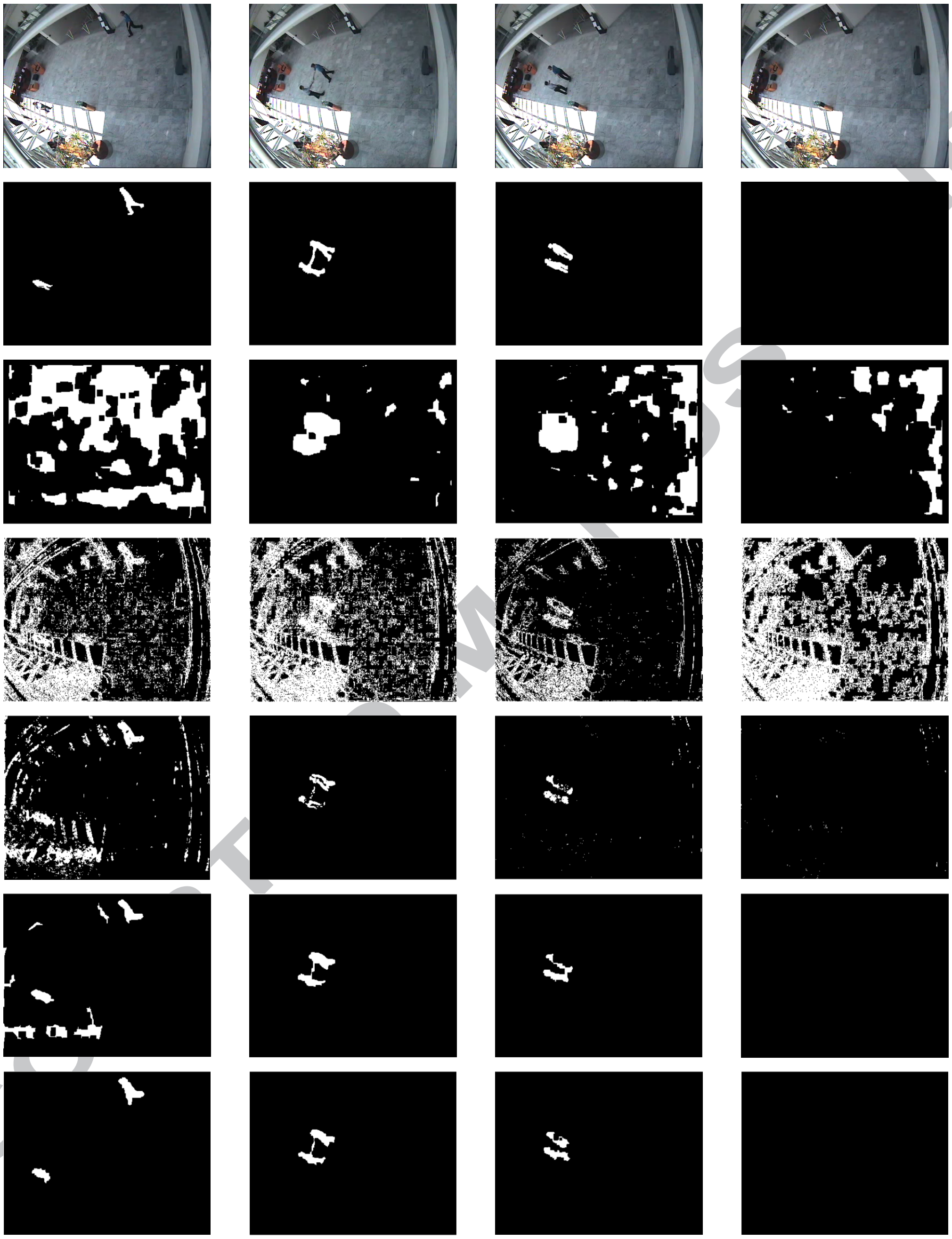
- *Video 4 [45]:* In this color indoor video sequence, two persons make entry into a hall from opposite directions with objects in their hands. There is not much variation in the color of the foreground and background. Variation in illumination can be noticed. There are total number of 299 frames with size 240 pixels × 352 pixels (i.e. row size × column size) in this video. The frame rate and bit rate of this video is 30 frames/sec and 24 kbps respectively.

Furthermore, we performed our experiment with MATLAB R2013a environment and executed on Intel (R) core (TM) i7-4770 cpu@3.40GHz processor with 4GB RAM. We have inspected the plot for T_1, the histogram for T_2, and also selected the parameter α depending on the signal to noise ratio. The final selected values of parameters T_1, T_2, and α for detecting the moving object for each video datasets are displayed in Table 1.

Table 1: Parameter value for video datasets

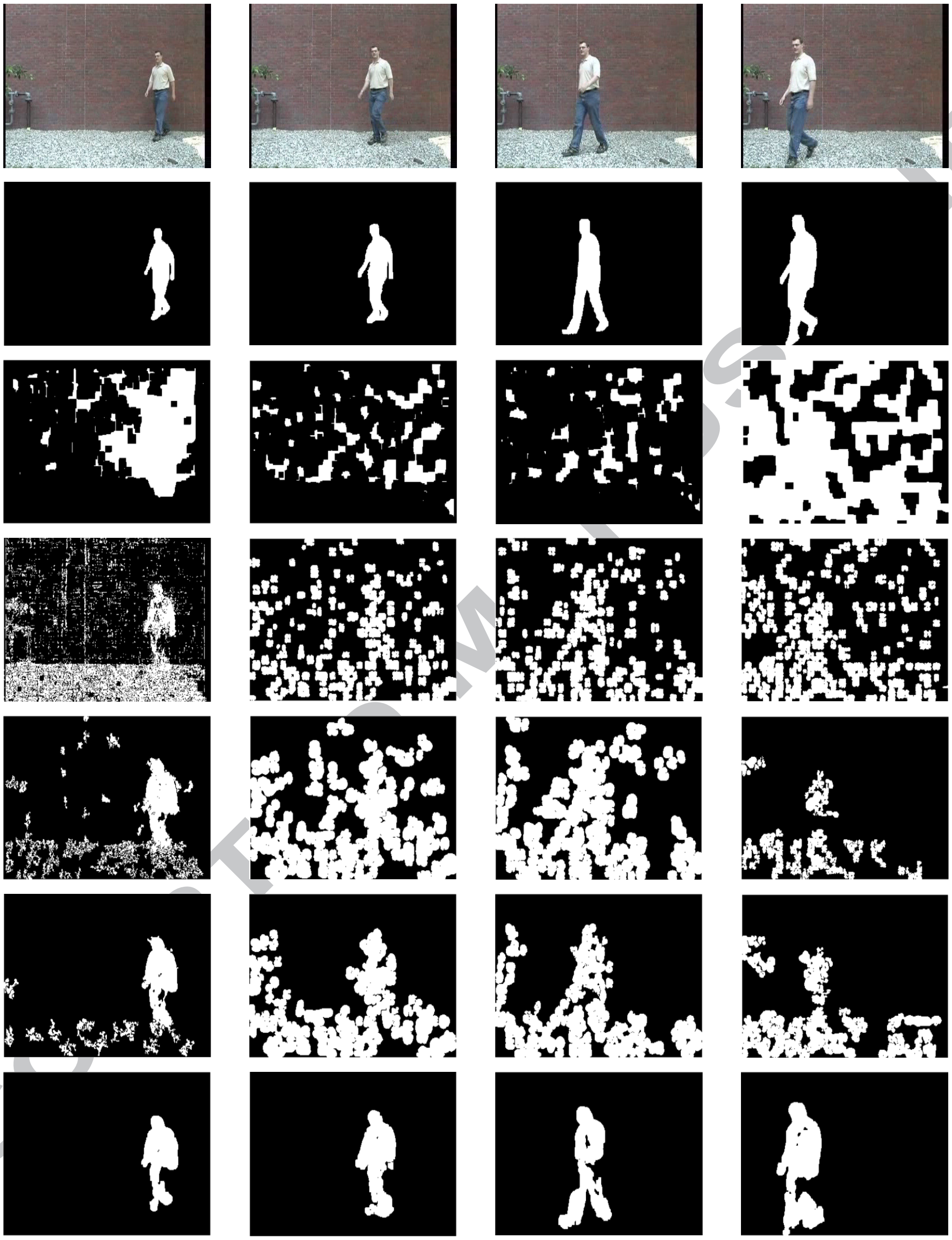
Input video	T_1	T_2	α	Block size
Video 1	90	2	30	8×8
Video 2	90	2	70	8×8
Video 3	90	4	10	8×8
Video 4	90	3	27	8×8

The results for some of the representative video frames are shown in Figs. 4–7. These figures display the original video frames, ground truth, and the foreground object obtained from the five methods of Video 1 (the frame number 264, 329, 362, 620), Video 2 (the frame number 4, 45, 92, 130), Video 3 (the frame number 4, 15, 33, 48) and Video 4 (the frame number 42, 72, 222, 265) respectively. The first two rows of the aforementioned figures display the original frames and their ground truth respectively. Next five rows (from top to bottom) of the Figs. 4–7 display the moving object detection result by LK [16], HS [29], Xin et al. [38], Sengar et al. [35], and proposed BBOF method. There are large number of false positives in all the video sequences for LK [16], HS [29], and Xin et al. [38] techniques (see rows 3, 4 and 5 of Fig. 4–7). The method proposed by Sengar et al. [35] also exhibits the false positives (see column a of row 6 of Fig. 4, row 6 of Fig. 5, columns a and b of row 6 of Fig. 6) as well as false negatives (see column c of row 6 of Fig. 4, column d of row 6 of Fig. 5, columns



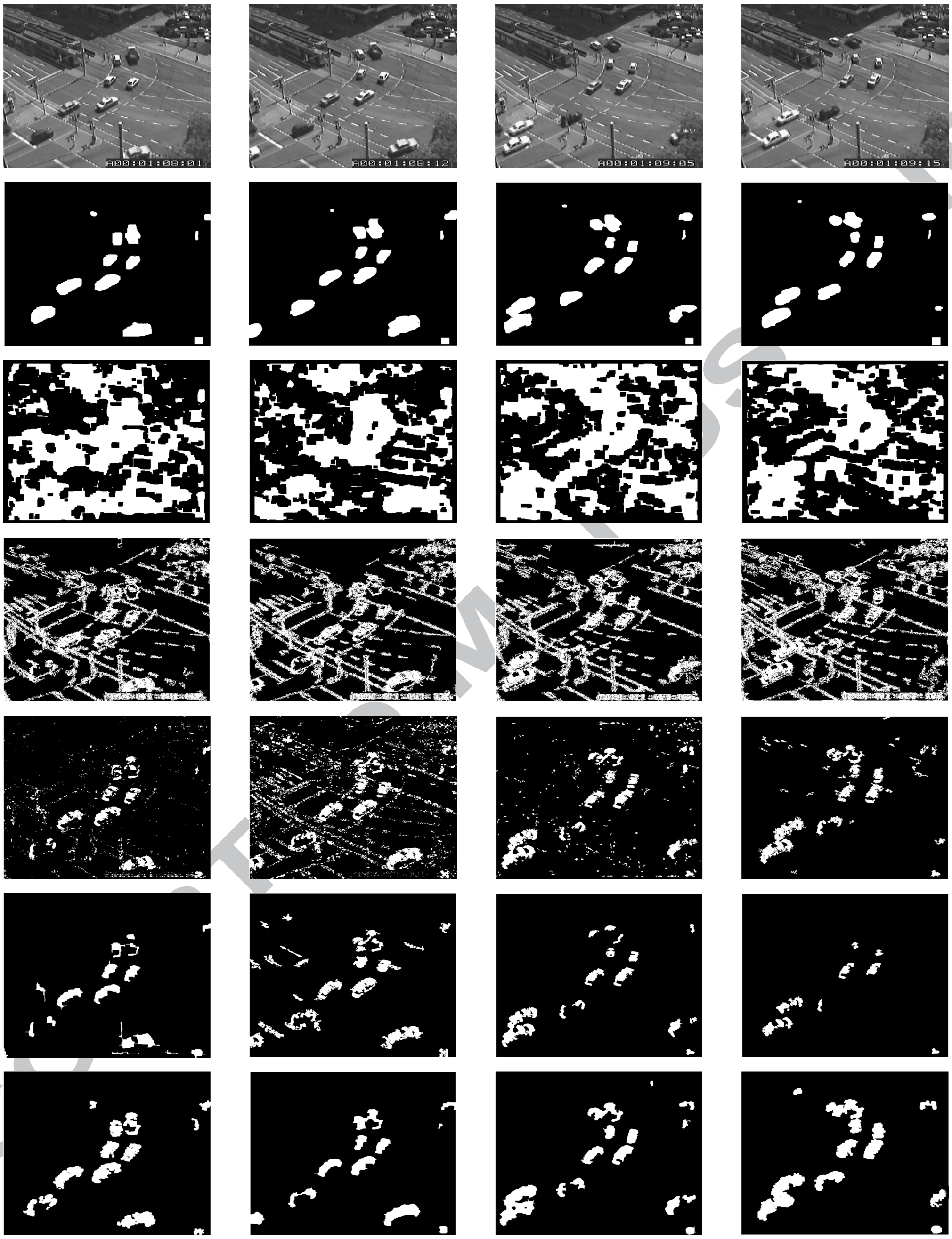
(a) Frame 264 (b) Frame 329 (c) Frame 362 (d) Frame 620

Figure 4: Results with Video 1 for the frames number (a) 264^{th} (b) 329^{th} (c) 362^{th} (d) 620^{th}; row wise, top to bottom: original frame, ground truth, LK [16], HS [29], Xin [38], Sengar [35], and BBOF.



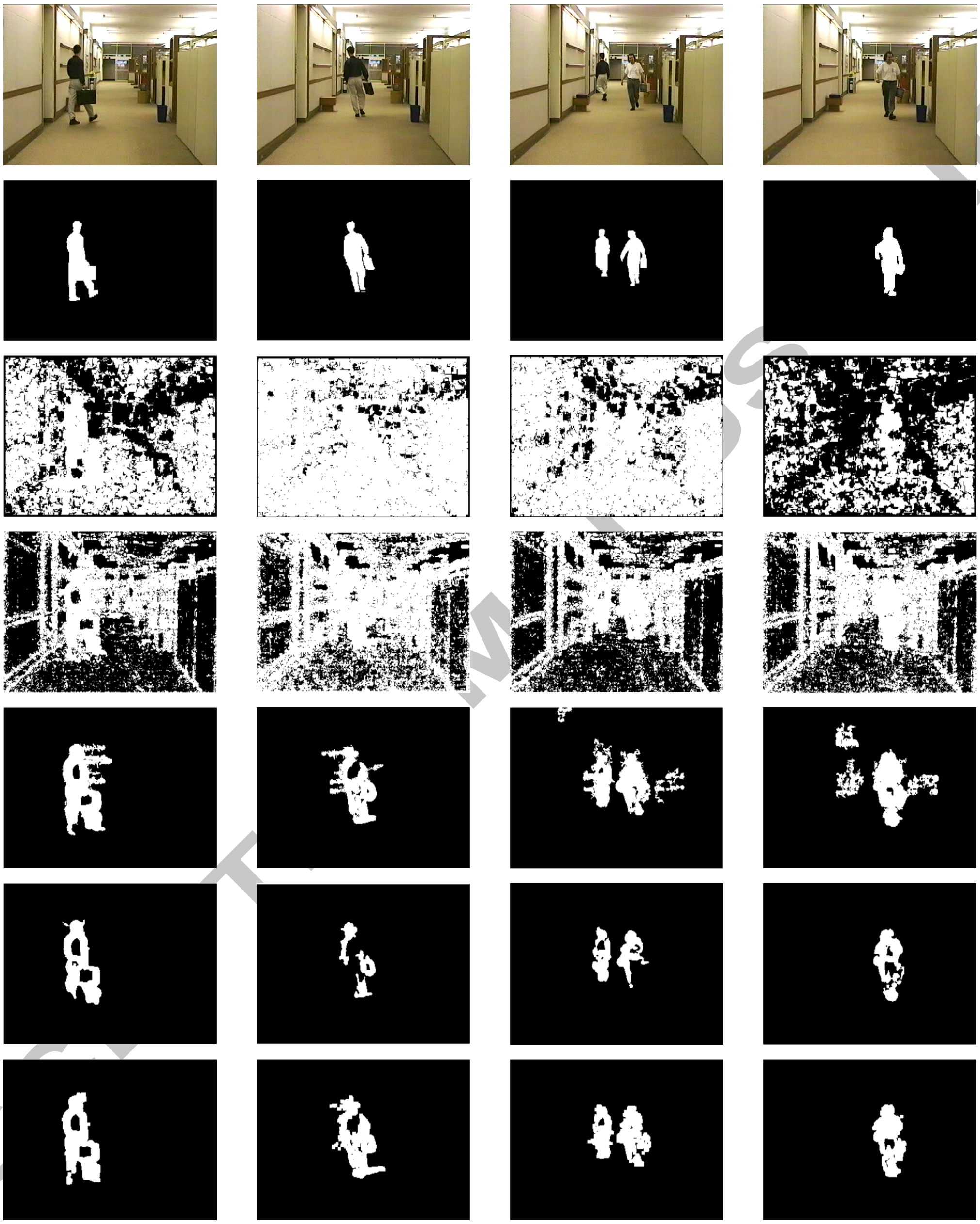
(a) Frame 4 (b) Frame 45 (c) Frame 92 (d) Frame 130

Figure 5: Results with Video 2 for the frames number (a) 4^{th} (b) 45^{th} (c) 92^{th} (d) 130^{th}; row wise, top to bottom: original frame, ground truth, LK [16], HS [29], Xin [38], Sengar [35], and BBOF.



(a) Frame 4 (b) Frame 15 (c) Frame 33 (d) Frame 48

Figure 6: Results with Video 3 for the frames number (a) 4^{th} (b) 15^{th} (c) 33^{th} (d) 48^{th}; row wise, top to bottom: original frame, ground truth, LK [16], HS [29], Xin [38], Sengar [35], and BBOF.



(a) Frame 42 (b) Frame 72 (c) Frame 222 (d) Frame 265

Figure 7: Results with Video 4 for the frames number (a) 42^{th} (b) 72^{th} (c) 222^{th} (d) 265^{th}; row wise, top to bottom: original frame, ground truth, LK [16], HS [29], Xin [38], Sengar [35], and BBOF.

a, c and d of row 6 of Fig. 6, columns b and c of row 6 of Fig. 7). As displayed in row 7 of Figs. 4–7, our BBOF method outperforms all the tested approaches, as there are less number of false positives as well as false negatives and thereby, our proposed technique has acquired reasonably high similarity with the ground truth and attains better integrity of foreground objects. For all the test video datasets, LK [16], HS [29], Xin et al. [38], Sengar et al. [35] techniques cannot correctly detect the foreground objects. Therefore, the resultant foreground objects cannot be isolated from the background pixels.

5.1. Performance evaluation

It is evident from the experimental results that our method outperforms other well known existing techniques. However, we have compared the performance of the proposed and other existing techniques at pixel-level and for that we consider motion detection as a binary classification. The performance of this classification is measured by means of the False Positive Rate (FPR), True Positive Rate (TPR), Precision and F-Score. These performance measuring parameters are defined as:

$$FPR = FP/(FP + TN) \tag{17}$$

$$TPR = TP/(TP + FN) \tag{18}$$

$$Precision = TP/(TP + FP) \tag{19}$$

$$F\text{–}Score = (2 \times Precision \times TPR)/(Precision + TPR) \tag{20}$$

Here TP (True Positive), FP (False Positive), FN (False Negative), and TN (True Negative) count the number of correctly classified foreground pixels, background pixels incorrectly detected as foreground, foreground pixels incorrectly detected as background, and correctly classified background pixels respectively.

To correctly detect the moving objects, the values of TPR, precision and F-Score should be high as well as FPR should be low. Tables 2–5 display the F-Score, Precision, FPR, and TPR of Video 1, Video 2, Video 3, and video 4 datasets respectively. Fig. 8 shows the bar charts of F-Score, Precision, FPR, and TPR values for the aforementioned datasets corresponding to Tables 2–5.

Table 2: F-Score of previously exist and proposed methods.

Input video	Method				
	LK [16]	HS [29]	Xin [38]	Sengar [35]	BBOF
Video 1	0.1226	0.0429	0.3029	0.4657	**0.6357**
Video 2	0.1855	0.1864	0.2217	0.3693	**0.7029**
Video 3	0.2900	0.2982	0.5808	0.5523	**0.7379**
Video 4	0.0926	0.1230	0.5331	0.5858	**0.6746**

Table 3: Precision of previously exist and proposed methods.

Input video	Method				
	LK [16]	HS [29]	Xin [38]	Sengar [35]	BBOF
Video 1	0.0703	0.0222	0.3180	0.5597	**0.6004**
Video 2	0.1283	0.1123	0.1418	0.2642	**0.6106**
Video 3	0.1734	0.1884	0.5966	**0.8074**	0.6998
Video 4	0.0490	0.0658	0.3887	**0.6034**	0.5910

LK [16], HS [29], and Xin et al. [38] techniques consist of high values of TPR as well as FPR for all the tested sequences, but for accurate result the values of TPR and FPR should be high and low respectively as is obtained in our BBOF method (see Tables 4, 5 and Fig. 8). Therefore, LK [16], HS [29], and Xin et al. [38] methods do not perform well with all the tested sequences. For the Video 2, Sengar et al. [35] misclassified large number of the background pixels as foreground which leads to high value of TPR as well as FPR and low value of precision as well as F-Score (see Tables 2–5 and Fig. 8(b)). Sengar et al. [35] cannot detect the whole foreground pixels for Video 3 and video 4, which results in low FPR and TPR (see Tables 4, 5 and Figs. 8(c), 8(d)). The precision value of Sengar et al. [35] is higher than the BBOF for Video 3 and Video 4 (see Table 3). However, the values of TPR and F-Score of our BBOF method are considerably high for these datasets (see Tables 2 and 5) and as given in Eq. 20, F-Score is the normalization of precision and TPR. Furthermore, the proposed

Table 4: FPR of previously exist and proposed methods.

Input video	Method				
	LK [16]	HS [29]	Xin [38]	Sengar [35]	BBOF
Video 1	0.1812	0.3265	0.0313	0.0209	*0.0076*
Video 2	0.3278	0.3491	0.3930	0.2249	*0.0553*
Video 3	0.3064	0.2112	0.0324	0.0086	*0.0230*
Video 4	0.6706	0.5267	0.0558	0.0148	*0.0254*

Table 5: TPR of previously exist and proposed methods.

Input video	Method				
	LK [16]	HS [29]	Xin [38]	Sengar [35]	BBOF
Video 1	0.9142	0.8865	0.6281	0.6174	*0.7389*
Video 2	0.5532	0.6691	0.8328	0.8124	*0.8650*
Video 3	0.9120	0.7191	0.5863	0.4355	*0.7842*
Video 4	0.8889	0.9648	0.8734	0.5803	*0.8528*

technique has considerably higher TPR, F-Score, and precision as well as lower FPR than Xin et al. [38], and Sengar et al. [35] for Video 1 and Video 2 datasets (see Tables 2–5 and Figs. 8(a), 8(b)). Here, The best results for F-Score and precision are displayed in **bold**. However, the best score for FPR and TPR cannot be shown individually because the TPR attains a high value for accurate detection while the FPR at the same time attains a low value. So if we collectively see the results of both FPR and TPR, then our technique performs better than the existing methods (see Tables 4 and 5). The best results for FPR and TPR are displayed in *italic.*

We have also shown the F-Score, precision, FPR and TPR for different techniques for randomly chosen frames of Video 1, Video 2, Video 3, and Video 4 in Figs. 9–12 respectively. Our technique outperforms the existing methods for most of the randomly selected frames i.e. the values of F-Score, precision and TPR are considerably higher as well as the value of FPR is lower than

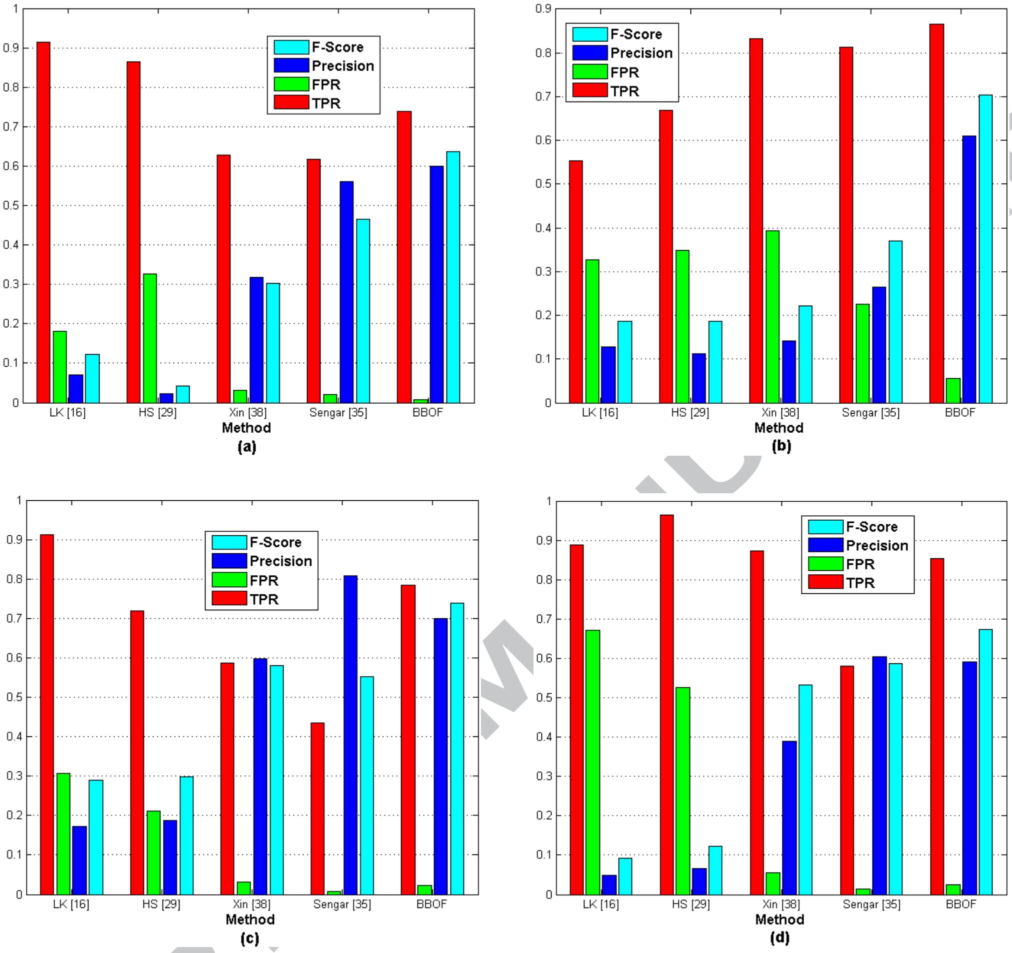


Figure 8: The bar charts showing the values of F-Score, Precision, FPR, and TPR for (a) Video 1; (b) Video 2; (c) Video 3; (d) Video 4.

those produced by other methods. However, in some cases, the values of TPR is high for existing techniques, but the value of FPR is also high, which is not good for accurate result (see LK and HS method in Fig. 9 and Fig. 12 as well as HS method in Fig. 11). For some frames of Video 3 and Video 4 (see Fig. 11 and Fig. 12), the value of precision for our BBOF method is lower than that of Sengar et al. [35], but the TPR and F-Score values of Sengar et al. [35] are very low for these sequences. Consequently our method is considerably better than

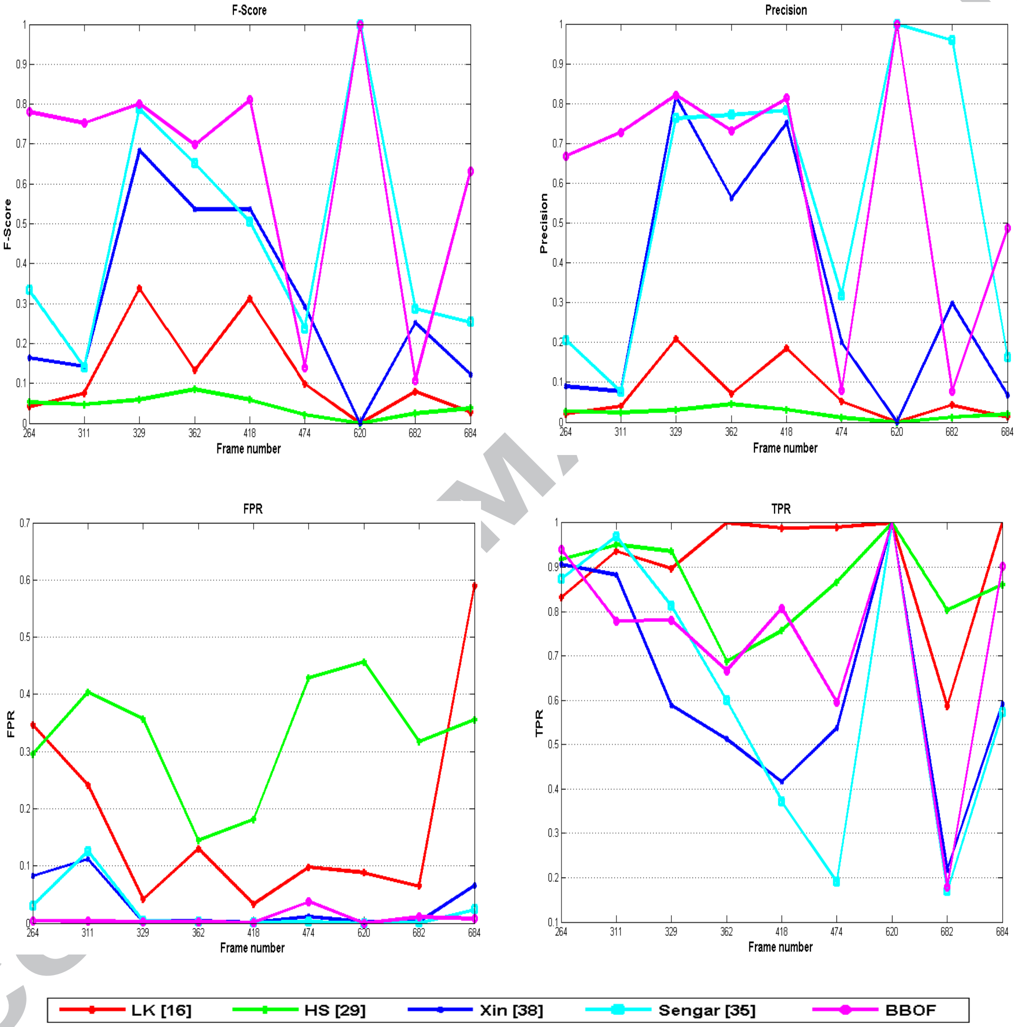


Figure 9: F-Score, Precision, FPR, TPR for different methods with Video 1.

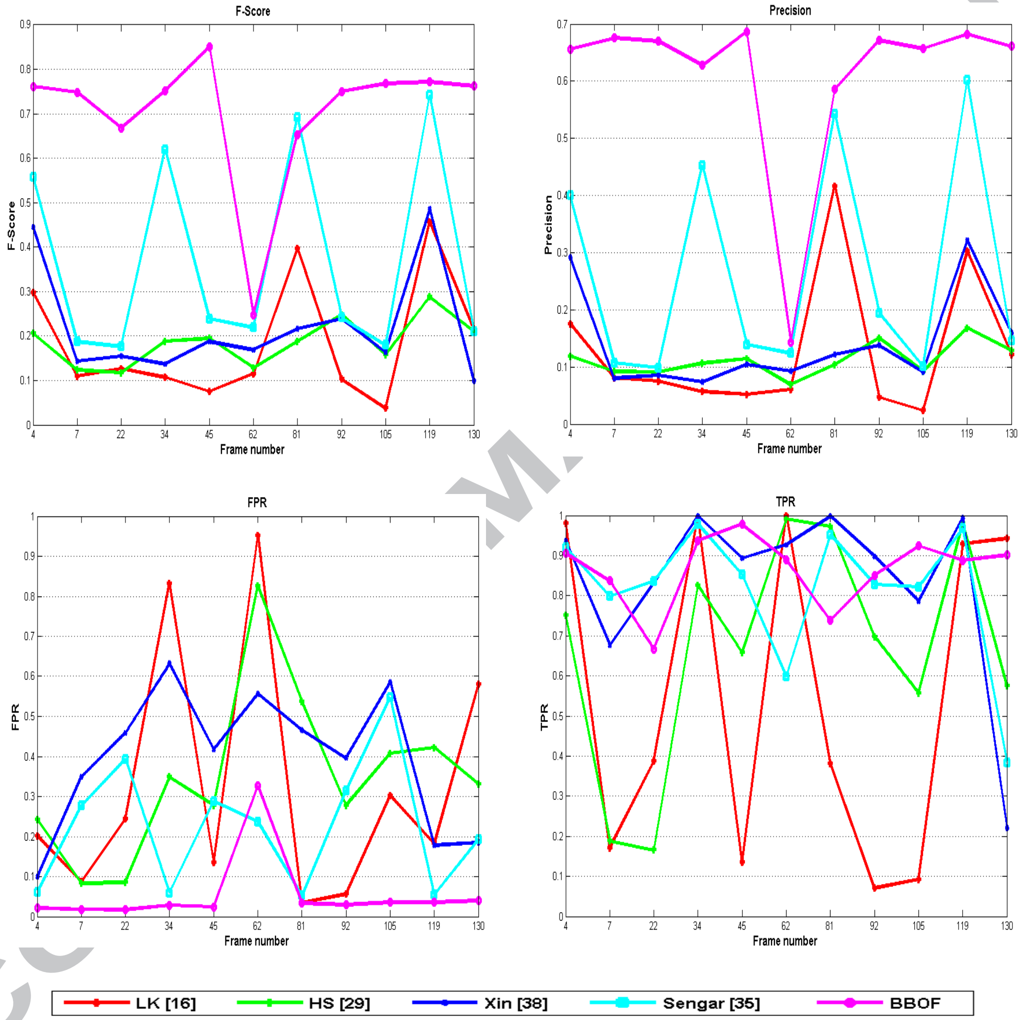


Figure 10: F-Score, Precision, FPR, TPR for different methods with Video 2.

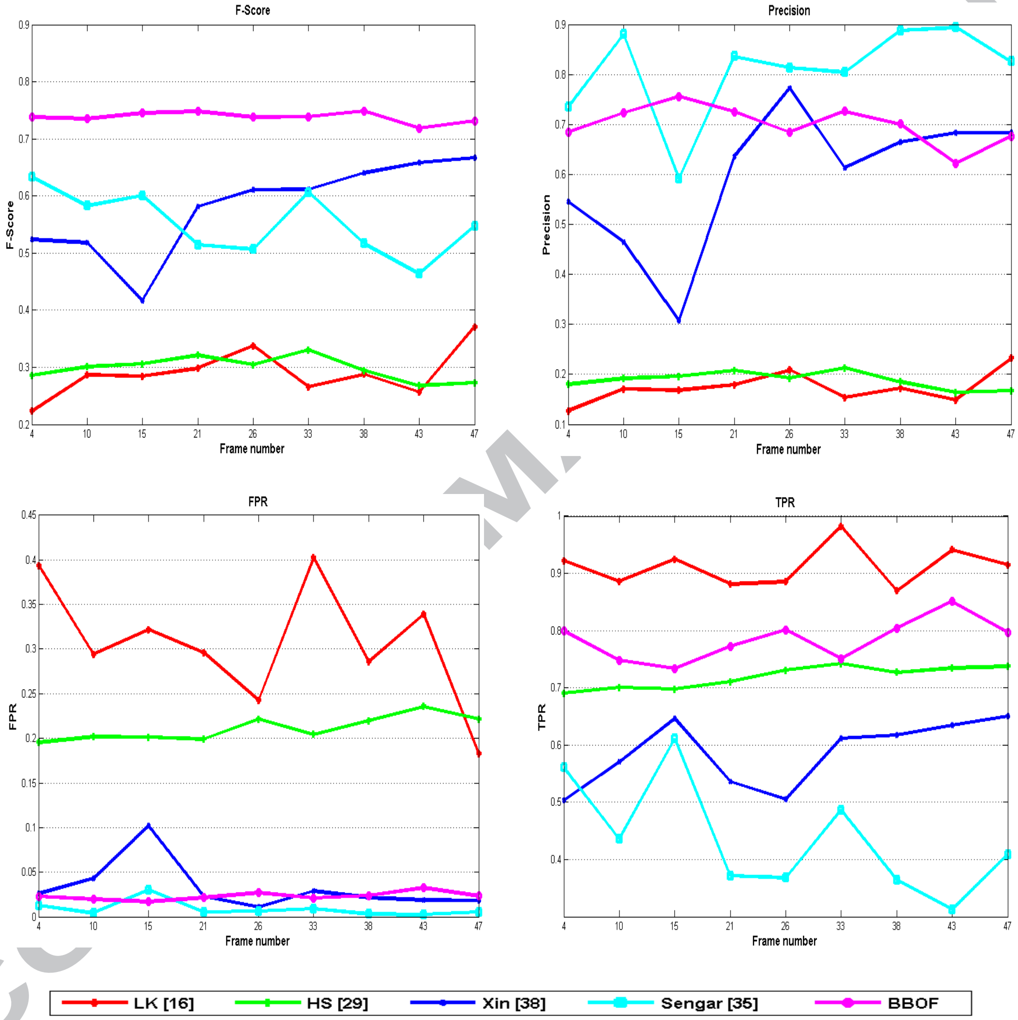


Figure 11: F-Score, Precision, FPR, TPR for different methods with Video 3.

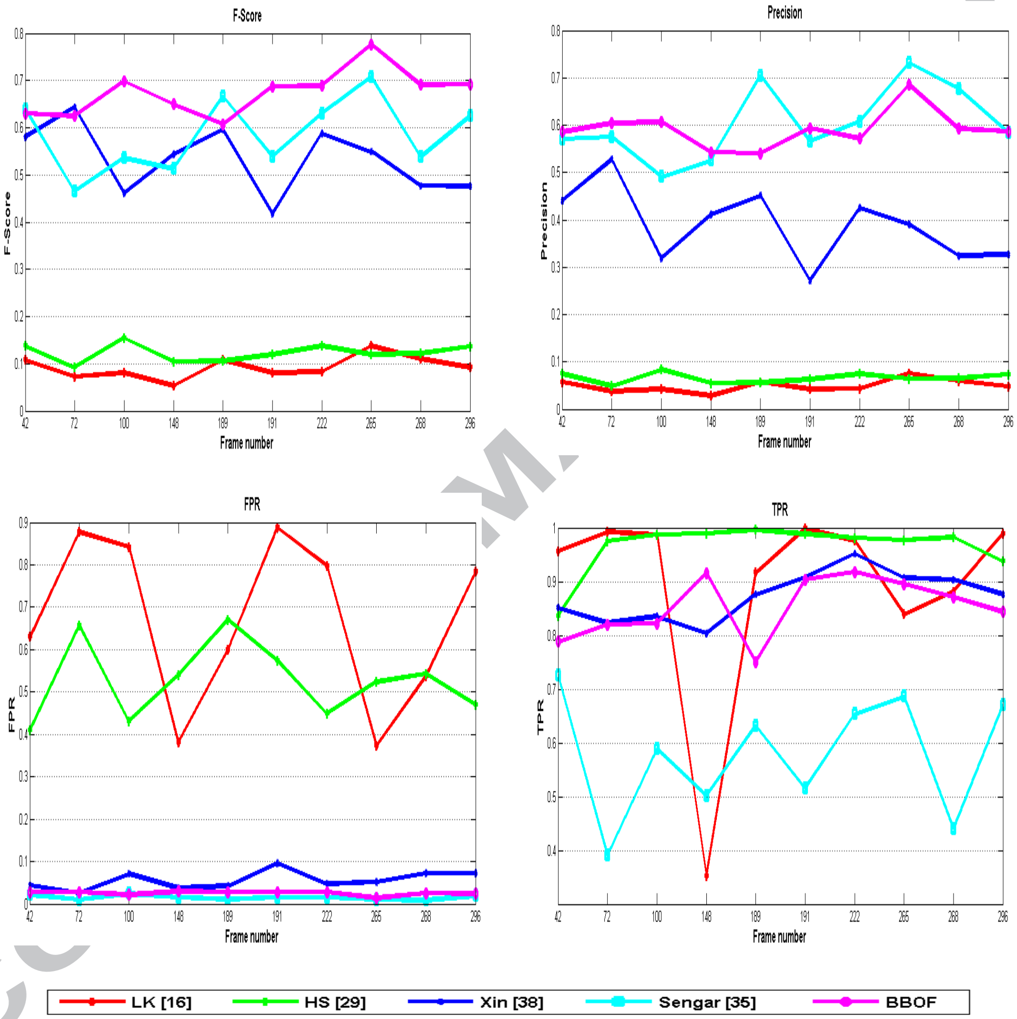


Figure 12: F-Score, Precision, FPR, TPR for different methods with Video 4.

Table 6: Average processing time/frame of existing and proposed methods in second.

Input video	Method				
	LK [16]	HS [29]	Xin [38]	Sengar [35]	BBOF
Video 1	3.77	0.87	2.15	0.95	**0.76**
Video 2	2.43	**0.72**	1.98	0.97	**0.72**
Video 3	5.84	1.78	5.04	1.96	**0.98**
Video 4	2.04	0.69	1.75	0.81	**0.51**

the existing techniques.

We have also done the performance evaluation of tested methods in terms of processing time. Table 6 depicts that existing techniques take longer time as compared to the proposed method (best values are displayed in **bold**). The overall performance evaluation proves that the proposed approach yields reasonably good results as compared to other approaches; detection accuracy of our approach is reasonably high, and the computation time is significantly low.

6. Conclusion

We analyzed the different methods for detecting moving objects and found that accuracy of detection is not satisfactory due to illumination variation and effect of noise etc. Therefore an approach for moving object detection from complex video datasets captured by a non-moving camera is presented. Our proposed approach detects almost all the foreground objects with less false alarm. This technique has enhancements compared to other existing approaches in terms of increased accuracy, and reduced processing time. Experimental results and performance evaluation show that our method can effectively handle the small and large size objects with slow and fast motion. In future, attempt will be made to enhance our method for moving camera as well.

References

[1] S. S. Sengar, S. Mukhopadhyay, Moving object tracking using Laplacian-DCT based perceptual hash, in: International Conference on Wireless Communications, Signal Processing and Networking, IEEE, 2016, pp. 2345–2349. doi:10.1109/WiSPNET.2016.7566561.

[2] N. Zhao, Y. Xia, C. Xu, X. Shi, Y. Liu, APPOS: An adaptive partial occlusion segmentation method for multiple vehicles tracking, Journal of Visual Communication and Image Representation 37 (2016) 25–31. doi:10.1016/j.jvcir.2015.04.011.

[3] O. Motlagh, D. Nakhaeinia, S. H. Tang, B. Karasfi, W. Khaksar, Automatic navigation of mobile robots in unknown environments, Journal of Neural Computing and Applications 24 (2014) 1569–1581. doi:10.1007/s00521-013-1393-z.

[4] J. Garcia, A. Gardel, I. Bravo, J. Lazaro, M. Martinez, D. Rodriguez, Directional people counter based on head tracking, IEEE Transactions on Industrial Electronics 60 (2013) 3991–4000. doi:10.1109/TIE.2012.2206330.

[5] S. Megrhi, M. Jmal, W. Souidene, A. Beghdadi, Spatio-temporal action localization and detection for human action recognition in big dataset, Journal of Visual Communication and Image Representation 41 (2016) 375–390. doi:10.1016/j.jvcir.2016.10.016.

[6] J. Candamo, M. Shreve, D. Goldgof, D. Sapper, R. Kasturi, Understanding transit scenes: a survey on human behavior-recognition algorithms, IEEE Transactions on Intelligent Transportation Systems 11 (2010) 206–224. doi:10.1109/TITS.2009.2030963.

[7] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, L. Wixson, A system for video surveillance and monitoring, Tech. Rep. CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University (May 2000).

[8] S. S. Sengar, S. Mukhopadhyay, A novel method for moving object detection based on block based frame differencing, in: Recent Advances in Information Technology (RAIT), 2016 3rd International Conference on, IEEE, 2016, pp. 467–472. doi:10.1109/RAIT.2016.7507946.

[9] T. Bouwmans, Traditional and recent approaches in background modeling for foreground detection: An overview, Computer Science Review 11 (2014) 31–66. doi:10.1016/j.cosrev.2014.04.001.

[10] S. S. Sengar, S. Mukhopadhyay, Detection of moving objects based on enhancement of optical flow, Optik-International Journal for Light and Electron Optics (2017) .doi:10.1016/j.ijleo.2017.07.040.

[11] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabaly, C. Quek, Video processing from electro-optical sensors for object detection and tracking in maritime environment: a survey, arXiv preprint arXiv:1611.05842 (2016) 1–23.

[12] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, E. H. Zahzah, Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset, Computer Science Review (2016) .doi:/10.1016/j.cosrev.2016.11.001.

[13] G. L. Foresti, C. Micheloni, C. Piciarelli, Detecting moving people in video streams, Pattern Recognition Letters 26 (14) (2005) 2232–2243. doi:10.1016/j.patrec.2005.03.031.

[14] A. Halidou, X. You, M. Hamidine, R. A. Etoundi, L. H. Diakite, Fast pedestrian detection based on region of interest and multi-block local binary pattern descriptors, Computers & Electrical Engineering 40 (8) (2014) 375–389. doi:10.1016/j.compeleceng.2014.10.003.

[15] A. F. Caballero, J. C. Castillo, J. M. Cantos, R. M. Tomas, Optical flow or image subtraction in human detection from infrared camera on mobile

robot, Journal of Robotics and Autonomous Systems 58 (2010) 1273–1281. doi:10.1016/j.robot.2010.06.002.

[16] J. Y. Bouguet, Pyramidal implementation of the affine Lucas kanade feature tracker description of the algorithm, Intel Corporation 5 (2001) 1–10.

[17] N. Paul, A. Singh, A. Midya, P. P. Roy, D. P. Dogra, Moving object detection using modified temporal differencing and local fuzzy thresholding, The Journal of Supercomputing (2016) 1–20doi:10.1007/s11227-016-1815-7.

[18] W. Hu, C. Chen, T. Chen, D. Huang, Z. Wu, Moving object detection and tracking from video captured by moving camera, Journal of Visual Communication and Image Representation 30 (2015) 164–180.

[19] T. Bouwmans, Background subtraction for visual surveillance: A fuzzy approach, in: S. Pal, A. Petrosino, L. Maddalena (Eds.), Handbook on soft computing for video surveillance, Taylor and Francis, 2012, Ch. 5, pp. 103–138.

[20] Y. Xu, J. Dong, B. Zhang, D. Xu, Background modeling methods in video analysis: A review and comparative evaluation, CAAI Transactions on Intelligence Technology 1 (1) (2016) 43–60.

[21] L. Maddalena, A. Petrosino, The SOBS algorithm: what are the limits?, in: Workshop on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 21–26. doi:10.1109/CVPRW.2012.6238922.

[22] N. M. Oliver, B. Rosario, A. P. Pentland, Bayesian computer vision system for modeling human interactions, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 831–843. doi:10.1109/34.868684.

[23] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: International Conference On Computer Vision and Pattern Recognition, IEEE, 1999. doi:10.1109/CVPR.1999.784637.

[24] I. Haritaoglu, D. Harwood, L. S. Davis, W^4: real-time surveillance of people and their activities, IEEE Transactions on pattern analysis and machine intelligence 22 (8) (2000) 809–830. doi:10.1109/34.868683.

[25] S. S. Sengar, S. Mukhopadhyay, Foreground detection via background subtraction and improved three-frame differencing, Arabian Journal for Science and Engineering 42 (8) (2017) 3621–3633. doi:10.1007/s13369-017-2672-2.

[26] J. V. D. Vyver, Detection of moving objects in the HEVC compressed domain for ultra-high resolution video, Master's thesis, Ghent University (June 2016).

[27] C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, F. Wu, Efficient parallel framework for HEVC motion estimation on many-core processors, IEEE Transactions on Circuits and Systems for Video Technology 24 (2014) 2077–2089. doi:10.1109/TCSVT.2014.2335852.

[28] C. Yan, Y. Zhang, J. Xu, F. Dai, L. Li, Q. Dai, F. Wu, A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors, IEEE Signal Processing Letters 21 (2014) 573–576. doi:10.1109/LSP.2014.2310494.

[29] B. K. P. Horn, B. G. Schunck, Determining optical flow, Journal of Artificial Intelligence 17 (1981) 185–203.

[30] E. Chen, X. Xu, X. Yang, W. Zhang, Quaternion based optical flow estimation for robust object tracking, Journal of Digital Signal Processing 23 (2013) 118–125. doi:10.1016/j.dsp.2012.07.017.

[31] L. Schwarz, A. Mkhitaryan, D. Mateus, N. Navab, Human skeleton tracking from depth data using geodesic distances and optical flow, Journal of Image and Vision Computing 30 (2012) 217–226. doi:10.1016/j.imavis.2011.12.001.

[32] I. H. Choi, J. M. Pak, C. K. Ahn, S. H. Lee, M. T. Lim, M. K. Song, Arbitration algorithm of FIR filter and optical flow based on ANFIS for visual object tracking, Journal of Measurement 75 (2015) 338–353. doi:10.1016/j.measurement.2015.07.020.

[33] J. Kim, G. Ye, D. Kim, Moving object detection under free-moving camera, in: 17^{th} IEEE International Conference on Image Processing, IEEE, 2010, pp. 4669–4672. doi:10.1109/ICIP.2010.5652848.

[34] M. Tagliasacchi, A genetic algorithm for optical flow estimation, Journal of Image and Vision Computing 25 (2007) 141–147. doi:10.1016/j.imavis.2006.01.021.

[35] S. S. Sengar, S. Mukhopadhyay, Moving object area detection using normalized self adaptive optical flow, Optik-International Journal for Light and Electron Optics 127 (16) (2016) 6258–6267. doi:10.1016/j.ijleo.2016.03.061.

[36] D. Liu, J. Yu, Otsu method and k-means, in: 9th International Conference on Hybrid Intelligent Systems, IEEE, 2009, pp. 344–349. doi:10.1109/HIS.2009.74.

[37] P. Liao, T. Chen, P. Chung, A fast algorithm for multilevel thresholding, Journal of Information Science and Engineering 17 (2001) 713–727.

[38] Y. Xin, J. Hou, L. Dong, L. Ding, A self-adaptive optical flow method for the moving object detection in the video sequences, International Journal for Light and Electron Optics 125 (19) (2014) 5690–5694. doi:10.1016/j.ijleo.2014.06.092.

[39] G. Deng, L. W. Cahill, An adaptive gaussian filter for noise reduction and edge detection, in: Nuclear Science Symposium and Medical Imaging, Vol. 3, IEEE, 1993, pp. 1615–1619. doi:10.1109/NSSMIC.1993.373563.

[40] W. Foy, Position-location solutions by Taylor-series estimation, IEEE Transactions on Aerospace and Electronic Systems 12 (1976) 187–194. doi:10.1109/TAES.1976.308294.

[41] E. Dougherty, R. Lotufo, Hands-on morphological image processing, Vol. 71, Washington: SPIE Optical Engineering Press, 2003. doi:10.1007/978-3-540-70932-9_13.

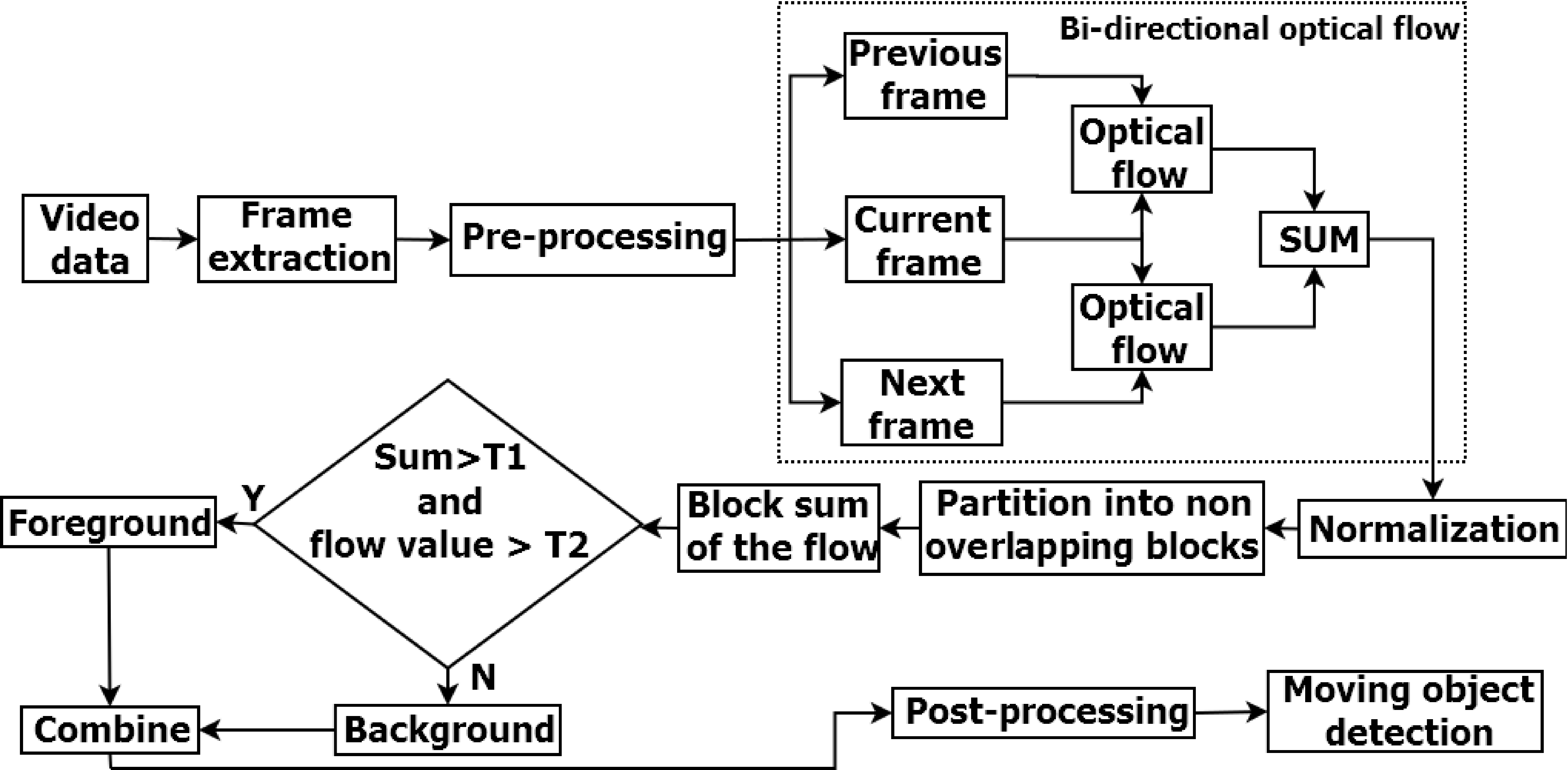
[42] Caviar test case scenarios, http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/, dataset, Dec, 2011.

[43] vidme, https://vid.me/videodata, videodata, July, 2015.

[44] S. S. Sengar, S. Mukhopadhyay, Moving object detection based on frame difference and w4, Signal, Image and Video Processing (2017) 1–8doi:10.1007/s11760-017-1093-8.

[45] Database: Images & video clips (2), http://see.xidian.edu.cn/vipsl/database_Video.html, collected by the HDTV group, july, 2006.

- Optical flow based moving object detection algorithm is proposed.
- Bi-directional optical flow field is employed for motion estimation and detection.
- A histogram and plot based thresholding mechanism is used for foreground detection.
- Moving objects are finally detected by employing morphological operation and connected component analysis on block based bi-directional optical flow field.
- The Proposed technique is tested and compared with existing approaches using real video datasets.


Bi-directional optical flow
Video data
Frame extraction
Pre-processing
Previous frame
Current frame
Next frame
Optical flow
Optical flow
SUM
Normalization
Partition into non overlapping blocks
Block sum of the flow
Sum>T1 and flow value > T2
Y
N
Foreground
Background
Combine
Post-processing
Moving object detection