International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

Moving Object Detection In Visual Surveillance Based On Background Subtraction

Asmita A. Kumbhar
Department of E&TC
TSSM's BSCOER, Pune

Ashwini M. Deshpande
Department of E&TC
TSSM's BSCOER, Pune

***Abstract*— Visual surveillance has attracted much attention in the computer vision community due to its potential applications. Background subtraction (BS) is a widely used approach for detecting moving objects in videos from static cameras. The Self-Organizing Background Subtraction (SOBS) algorithm implements an approach to moving object detection based on the neural network background model automatically generated by a self-organizing method, without prior knowledge about the involved patterns. The SOBS can handle scenes containing moving backgrounds, gradual illumination variations, and camouflage. In this paper performance of BS and SOBS methods is compared in terms of accuracy and processing speed for video sequences.**

Index Terms*— Motion Detection, Background Subtraction, Neural Network, Self Organizing map, Visual surveillance.

I. INTRODUCTION

Surveillance is the monitoring of behavior. Surveillance is the process of monitoring the behaviour of people, objects or processes within systems for conformity to expected or desired norms in trusted systems for security or social control. Visual surveillance has attracted much attention in the computer vision community due to its potential applications.

In video surveillance system, human operator responsible for monitoring does all task while watching the videos coming from the different cameras. Its a tedious job of an operator to watch the multiple screen and at the same time to be vigilant from any unfortunate event. These systems are ineffective for large crowded places as the number of cameras exceeds the capability of human experts. Such systems are used in widely used across the world.

Therefore the aim of visual surveillance is not only to put cameras in the place of human eyes, but also to accomplish the entire surveillance task as automatically as possible.

The usual approach to moving object detection is through background subtraction that consists in maintaining an up-to date model of the background and detecting moving objects as those that deviate from such a model. The problem with background subtraction is to automatically update the background from the incoming video frame and it should be able to overcome the following problems [1]:

• Illumination changes: The background model should be able to adapt, to gradual changes in illumination over a period of time.

• Moving background: Non-stationary background regions, such as branches and leaves of trees, a flag waving in the wind, or flowing water, should be identified as part of the background.

• Shadows: Shadows cast by moving object should be identified as part of the background and not foreground.

• Bootstrapping: The background model should be able to maintain background even in the absence of training background (absence of foreground object).

• Camouflage: Moving object should be detected even if pixel characteristics are similar to those of the background.

An adaptive SOBS model tries to overcome the problems of BS method. This paper compares the traditional BS and a new SOBS method for object tracking in video sequences on the basis of detection accuracy and processing speed.

The paper is organized as follows. In Section II, we present overview of existing approaches adopted for background subtraction. Section III reports the motion detection methods. In Section IV, tracking system is discussed. In Section V, consist of experimental evaluation of BS and SOBS methods. Section VI includes conclusions and its related discussion.

II. RELATED WORK

Due to its pervasiveness in various contexts, background subtraction has been afforded by several researchers, and plenty of literature has been published. There is no unique classification of proposed methods. There are some advantages of existing methods which include the following.

• **Parametric versus nonparametric**: Parametric models are tightly coupled with underlying assumptions, not always perfectly corresponding to the real data, and the choice of parameters can be cumbersome, thus reducing automation. On the other hand, nonparametric models are more flexible but heavily data dependent [2].

• **Unimodal versus multimodal**: Basic background models assume that the intensity values of a pixel can be modeled by a single unimodal distribution. Such models usually have low complexity, but cannot handle moving

backgrounds, while this is possible with multimodal models at the price of higher complexity [2].

•**Recursive versus nonrecursive**: Nonrecursive techniques store a buffer of a certain number of previous sequence frames and estimate the background model based on the temporal variation of each pixel within the buffer, while recursive techniques and recursively update a single background model based on each input frame [3].

• **Region-based versus pixel-based**: The region- based methods take advantage of inter pixel relations, segmenting the images into regions or refining the low-level classification obtained at the pixel level while pixel-based methods consist of the time series of observations is independent at each pixel [3].

Our implementation approach is based on the pixel based background subtraction method. The region growing approach is used for tracking system.

III. Motion Detection

Visual surveillance system include task such as motion detection, object classification, personal identification, tracking, and activity recognition. Out of the tasks mentioned above, detection of moving object is the first important step and successful segmentation of moving foreground object from the background ensures ease and accuracy of further step.

Motion detection or segmentation in image sequences aims at detecting regions corresponding to moving objects. Detecting moving regions provides information about later processes such as tracking and behavior analysis because only these regions need be considered in the later processes. Hu et al. [4] categorized motion detection into three major classes of method as frame differencing, optical flow, and background subtraction. Our approach is based on the background subtraction method.

A) Background subtraction

The background subtraction method is the common method of motion detection. It is a technology that uses the difference of the current image and the background image to detect the motion region, and provide object information. This method starts with the initialization and update of the background image. The effectiveness of both will affect the accuracy of test results. Therefore, this paper uses an effective method to initialize the background, and update the background in real time. The flow chart for moving human body extraction mechanism is shown in Fig. 1 [5].

1) Background image initialization

There are many ways to obtain the initial background image. For background image initialization, the first frame is treated as the background directly, or the average pixel brightness of the first few frames are considered as background.

2) Background Update

Background subtraction cannot handle illumination variation and results in noise in the motion detection mask. The problem of noise can be overcome, if the background is updated in every frame. In detection of the moving object, the pixels judged as belonging to the moving object maintain the original background gray values, not be updated. To update the background model morphological operation 'bwareaopen' is used in this paper. BW2 = bwareaopen(BW,P) removes from a binary image all connected components (objects) that have fewer than P pixels, producing another binary image, BW2. The value of P is taken 100 by experimental trials. This removes object smaller than 100 pixel and background get updated.

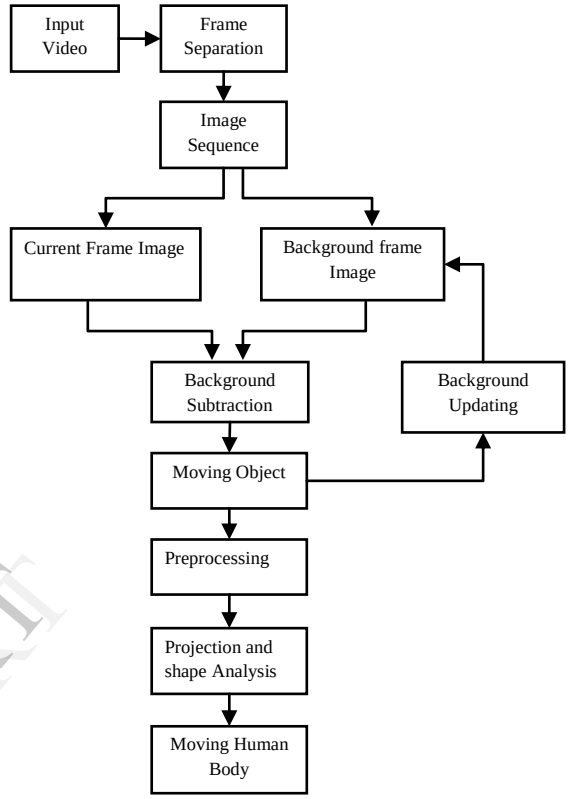

Figure1. The flow chart of moving human body extraction

3) Moving Object Extraction

After the background image $B(x, y)$ is obtained, subtract the background image $B(x,y)$ from the current frame $F_k(x, y)$. If the pixel difference is greater than the set threshold T, then that pixel will be considered as moving object, otherwise, the pixel considered as background pixels. The moving object is detected after threshold operation. Its expression is as follows [5]:

$$D_k(x,y) = \begin{cases} 1 & |F_k(x,y) - B_{k-1}(x,y)| > T \\ 0 & others \end{cases} \quad \ldots\ldots (1)$$

where the binary image $D_k(x,y)$ of differential results, T is gray-scale threshold and its value is decided as 5 by experimentation.

4) Pre-processing

The difference image obtained in moving object extraction step contains the motion region, consist of a large noise. Therefore, noise needs to be removed. For this purpose morphological methods are used on this binary image and noise is removed.

5) Extraction of Moving Human Body

After pre-processing, some accurate edge regions will be available, but the region related to the moving human body could not be determined. Extraction of moving human body is done by shape analysis and tracking process.

B) Self Organizing Background subtraction

The drawback of the background subtraction method to moving object detection is its extreme sensitivity to dynamic scene changes due to lighting and extraneous events. The idea is to build the background model by learning in a self-organizing manner many background variations. Based on the learnt background model through a map of motion and stationary patterns, this implementation can detect motion and selectively update the background model. Each node computes a function of the weighted linear combination of incoming inputs, where weights resemble the neural network learning. Each node can be represented by a weight vector, obtained collecting the weights related to incoming links. The set of weight vectors is called a model. An incoming pattern is mapped to the node whose model is "most similar" to the pattern, and weight vectors in a neighborhood of such node are updated. Therefore, the network behaves as a competitive neural network that implements a winner take-all function with an associated mechanism that modifies the local synaptic plasticity of the neurons, allowing learning to be restricted spatially to the local neighborhood of the most active neurons [1].

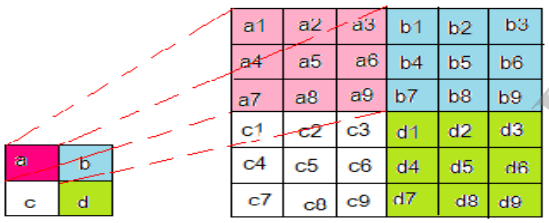

Figure 2. (Left) Simple image and the (right) neuronal map structure

For each color pixel, it considers a neuronal map consisting of n×n weight vectors. Each incoming sample is mapped to the weight vector that is closest according to a suitable distance measure, and the weight vectors in its neighborhood are updated. The whole set of weight vectors acts as a background model that is used for background subtraction in order to identify moving pixels.

1) Initial Background Model

In order to represent each weight vector, we choose the HSV color space, relying on the hue, saturation and value properties of each color. Such color space allows us to specify colors in a way that is close to human experience of colors. Let (*h, s, v*) be the HSV components of the generic pixel (*x, y*) of the first sequence frame I_0, and let C= (c_1, $c_2 \ldots c_n^2$) be the model for pixel *(x, y)*. Each weight vector c_i, i=1,2……, n^2 is a 3-D vector initialized as $c_i = (h, s, v)$.

The complete set of weight vectors for all pixels of an image *I* with *N* rows and *M* columns is represented as a neuronal map *A* with $n \times N$ rows and $n \times M$ columns, where the weight vectors for the generic pixel *(x, y)* of *I* are at neuronal map positions *(i,j)*, where $i=n \times x,\ldots\ldots,n\times(x+1)$-1 and $j=n\times y,\ldots.,n\times(y+1)$-1. An example of such neuronal map structure for a simple image *I* with *N=2* rows and *M=3* columns obtained choosing *n=3* is given in Fig.2. As depicted, the upper left pixel *a* of *I* (Fig. 2, left) has weight vectors ($a_1,\ldots\ldots,a_9$) stored into the 3 x 3 elements of the upper left part of neuronal map (Fig. 2, right) [1].

2) Subtraction and Update of the Background Model

After initialization, temporally subsequent samples are fed to the network. Each incoming pixel P_t of the t^{th} sequence frame I_t is compared to the current pixel model *C* to determine if there exists a weight vector that best matches it. If a best matching weight vector c_m is found, it means that P_t belongs to the background and it is used as the pixel encoding approximation, and the best matching Weight vector, together with its neighbourhood, is reinforced. In the latter case P_t is detected as belonging to a moving object (foreground) [1].

i. Finding the best match in to current Sample:

To find which weight vector gives the best match, several metrics for detecting changes in color imagery could be adopted .To find best match Euclidean distance of vectors in the HSV color hexcone ,that gives the distance between two pixels $P_p=(h_p, s_p, v_p)$ and $P_q=(h_q, s_q, v_q)$ as

$$d(I_t(p), I_t(p)) = \|(v_p s_p cos(h_p), v_p s_p sin(h_p), v_p) - v_q s_q cos(h_q), v_q s_q sin(h_q), v_q\|_2^2 \quad .. (2)$$

Indeed, the representation of HSV values as vectors in the HSV color hexcone used in such distance measure allows to avoid problems connected with the periodicity of hue (that represents an angle) and with the instability of hue for small values of saturation (hue is undefined for null saturation)

Weight vector c_m, for some m, gives the best match for the incoming pixel P_t if its distance from is minimum in the model *C* of P_t and is no greater than a fixed threshold
i=1,..,n2

$$d(c_m, p_t) = \min_{i,j=0,\ldots,n-1} d(c_i, p_t) \leq \epsilon, \ldots\ldots. (3)$$

The threshold ϵ gives the difference between foreground and background pixels, and is chosen as,

$$\epsilon = \begin{cases} \epsilon_1 & if\ 0 \leq t \leq K \\ \epsilon_2 & if\ t > K \end{cases} \quad \ldots\ldots\ldots (4)$$

with ϵ_1 and ϵ_2 small constants. Specifically, ϵ_1 should be greater than ϵ_2, since higher values for ϵ_1 allow, within the first K sequence frames, to obtain a background model including several observed pixel intensity variations, while lower values for ϵ_2 allow to obtain a more accurate background model in the online phase [1].

ii. Updating in the neighborhood of c_m:

If a best match c_m is found for current sample, Pt the weight vectors in the $n \times n$ neighborhood of c_m are updated according to selective weighted running average. Given the incoming pixel P_t *(x ,y)* at spatial position *(x, y)* and time *t* , if there exists a best match c_m in the model *C* of P_t , and c_m is present in the background model at position *(x, y)*, then weight vectors in the neighborhood of are updated according to-

$$A_t(i,j) = \left(1 - \propto_{i,j}(t)\right) A_{t-1}(i,j) + \propto_{i,j}(t) p_t(x,y) \quad \text{.. (5)}$$

In $\propto_{i,j}(t) = \propto(t) w_{i,j}$, where $w_{i,j}$ are Gaussian weights in the neighborhood, that well correspond to the lateral inhibition activity of neurons. Moreover, *α(t)* represents the learning factor, chosen as

$$\propto(t) = \begin{cases} \propto_1 - t\frac{\propto_1 - \propto_2}{K}, & if\ 0 < t \leq K \\ \propto_2, & if\ t > K \end{cases} \quad \text{..... (6)}$$

where $\alpha_1$and α_2 are predefined constants such that $\alpha_2 \leq \alpha_1$.

IV. TRACKING SYSTEM

Detected objects are identified and analyzed through different blobs. Then tracking is performed by matching corresponding features of blob. For this purpose Angular Deviation of Center of Gravity (ADCG) method is used [6]. For calculation of features, image blob is obtained on detected object. Each blob a feature vector is calculated which consists of

1) size of blob
2) center coordinates of blob
3) average color of blob

The matching of blob is done in sequence of frames for tracking by calculating Euclidean distance between two frames F1 and F2 for each pair of blob's feature as

$$\text{Dist} = \sqrt{\Sigma\ [\![(F1\text{-}F2)]\!]} \quad \text{..... (7)}$$

The blob pair with minimum distance is considered as tracked pair and other pairs are discarded for that corresponding blob. This process is continued for complete video and thus tracking of multiple persons is achieved [5].

V. EXPERIMENTAL EVALUATION

The BS and SOBS described in previous section are verified experimentally. For experimental evaluation 5 different video sequences are captured in different illumination conditions. For comparison two video sequences with resolution 179 x 320 pixels are considered here. Video sequence1 is taken at 484 frames per second. The video sequence 2 is taken at 768 frames per second.

In this, video is given as the input and that video are converted into frames. The main goal of the background subtraction is to detect the moving objects from the video sequence taken from the stationary camera.

A. Results for background subtraction method:

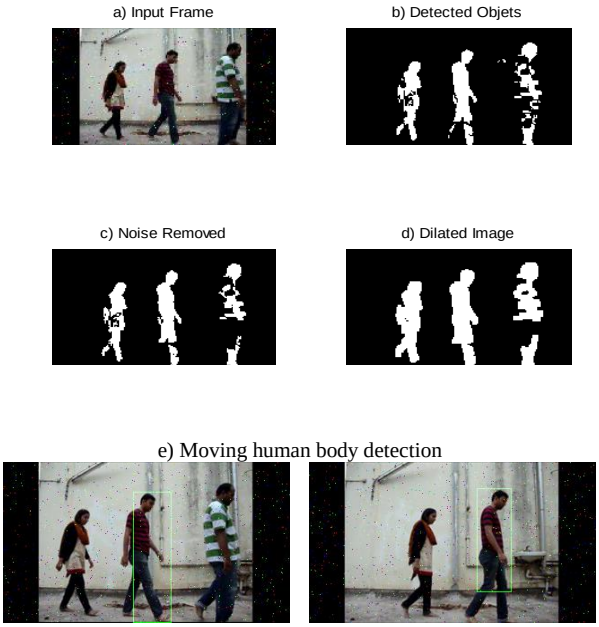

Figure 3: Video sequence 1- a) Input frame
b) Detected object c) Noise removal step
d) Dilated image e) Moving human body detection

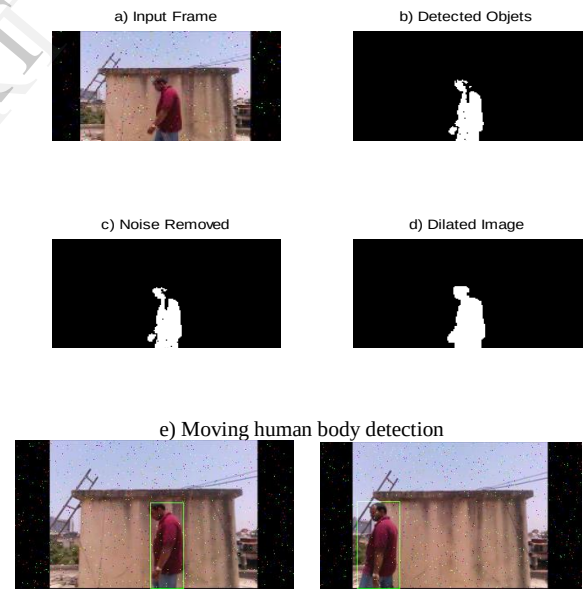

Figure 4: Video sequence 2: a) Input frame
b) Detected object c) Noise removal step
d) Dilated image e) Moving human body detection

B. Results for self organizing background subtraction method:

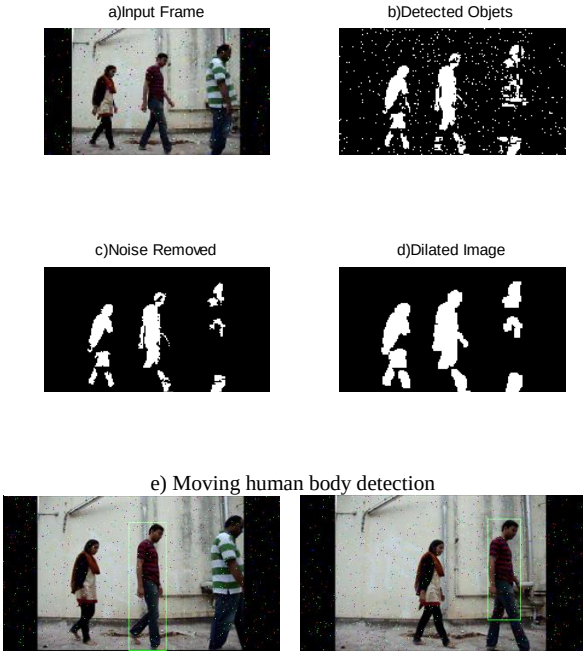

Figure 5: Video sequence 1 a) Input frame
b) Detected object c) Noise removal step
d) Dilated image e) Moving human body detection

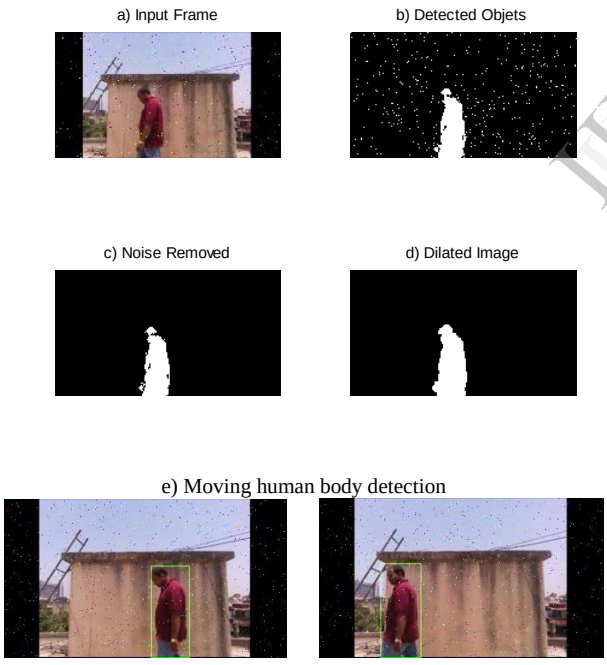

Figure 6: Video sequence 2 a) Input frame
b) Detected object c) Noise removal step
d) Dilated image e) Moving human body detection

Video sequence 1 is captured in low illumination, whereas video sequence 2 is captured in high illumination. Fig. 3 and Fig. 4 show the experimental results for BS. Fig. 5 and Fig. 6 show the results for SOBS. A fixed threshold method is used for background subtraction in both cases.

In background subtraction method, processing time required for video sequence 1 and video sequence 2 are 117.1809s and 244.92s respectively. In SOBS, processing time required for video sequence 1 and video sequence 2 are 501.8321s and 1.095e+003s respectively.

SOBS has very good adaptability in the high and low illumination environment. It also has a very good effect on the elimination of noise and be able to extract the complete and accurate picture of moving human body.

VI.CONCLUSION

In this paper object detection using basic background subtraction method and self organizing background subtraction method, which is based on neuronal mapping scheme, are discussed. BS gives very good results when there is static background while SOBS gives good results when there is dynamic background. BS is faster than SOBS when compared on the basis of processing time. The SOBS method is very effective when there is a gradual illumination change, cast shadows etc.

REFERENCES

[1] Lucia Maddalena and Alfredo Petrosino "A Self Organizing Approach to Background Subtraction for Visual Surveillance Applications." IEEE Transactions, VOL.17, NO.7, JULY 2008 .

[2] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, May 1997.

[3] B. P. L. Lo and S. A. Velastin, "Automatic congestion detection system for underground platforms," in *Proc. ISIMP*, 2001, pp. 158–161.

[4] Weiming Hu, Tieniu Tan, Liang Wang, and S.Maybank. " A Survey on Visual Surveillance of Object Motion and Behaviors". IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 34(3):334–352, August 2004.

[5] Lijing Zhang and Yingli Liang, "*Motion human detection based on background Subtraction*'' Second International Workshop on Education Technology and Computer Science, 284-287,Year: 2010.

[6] Debmalya Sinha, Gautam Sanyal, "Development of Human Tracking System For Video Surveillance", David Bracewell, et al. (Eds): AIAA 2011, CS & IT 03, pp. 187–195, 2011.

[7] Tao Jianguo, Yu Changhong, "Real-Time Detection and Tracking of Moving Object",Intelligent Information Technology Application, 2008.

[8] Weiming Hu, Tieniu Tan, Liang Wang, and S.Maybank. "A Survey on Visual Surveillance of Object Motion and Behaviors". IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 34(3):334–3512, August 2004.

[9] Lucia Maddalena and Alfredo Petrosino "*The SOBS algorithm: what are the limits*?" IEEE 2012 .