NOVEL FRAMEWORK FOR VIDEO OBJECT SEGMENTATION AND TRACKING USING AUTOMATIC THRESHOLD DECISION AND DIFFUSION DISTANCE

Elavarasan T #1, Vidhya R*2

# P.G Scholar
Department of Computer Science and Engineering
Nandha College of Technology
Erode, Tamilnadu.

*Assistant Professor
Department of Computer Science and Engineering

1E-mail: ctelavarasan@gmail.com
2E-mail: vidyaprathi@gmail.com

Abstract: Video surveillance process takes video as an input, processes the video frames and performs actions accordingly. The video surveillance process consists of many phases. Video object segmentation and tracking are two crucial building blocks of smart surveillance systems. Threshold decision is a complex problem for video object segmentation with a multibackground model. Some conditions like nonrigid object movement, target appearance variations due to changes in illumination, and background mess make robust video object tracking complex. Multiple thresholds might be needed in connection with more refined algorithms. To make it completely automatic for variant conditions, First an automatic threshold decision technique that can automatically and specifically determine the threshold values for dynamic backgrounds is proposed. Second, a video object tracking framework based on a particle filter is proposed with the probability function composed of diffusion distance for measuring color histogram similarity and motion clue from video object segmentation. The proposed framework will track multiple moving objects under drastic changes in illumination and background mess.

1 Introduction

Content analysis in smart surveillance has become progressively more important, and emerging systems [1]–[3] are being studied and deployed in real environments. In future surveillance networks, content analysis engines embedded in smart cameras will play an important role [4]. In embedded content analysis algorithms, video object segmentation and tracking get the most attention as they are crucial building blocks for other smart surveillance functions. In [5]–[8], several simple and efficient video object segmentation algorithms are proposed. However, the proposed algorithms cannot address dynamic backgrounds because only one background layer is employed in their background model. However, in [9]–[11], multilayered, complex background models are employed to address dynamic backgrounds; the large memory requirements of these models result in implementation bottlenecks, especially in embedded systems, such as smart cameras [12]. Performance comparisons of most existing video object segmentation algorithms are conducted with artificial datasets [13] that indicate that algorithms with multilayer background models (multimodels) outperform those with single background models (unimodels).

A threshold decision algorithm for deciding appropriate threshold values is also very important since the segmentation results deeply rely on the threshold value. In [15], a threshold decision method for single background models was proposed; however, it still cannot successfully and robustly determine the threshold values for conditions with dynamic background.

For video object tracking, the data association of segmentation blobs [16] is highly dependent on the feature of segmentation results. Gradient descent-based methods [17], [18] search for the most likely object candidate regions with
gradient descent optimization techniques. However, they suffer from local minima problems, and it is difficult for them to address objects that have large motions. In [22], the Kalman filter is employed to predict object motion and track objects; however, it may fail for objects that have random motions. Particle filter [23], [24] is a more robust methodology for object tracking, and it can address large and random motions more effectively; however, the features employed for object modeling and the distance measurements used to decide the weights of the particles, which are essential for making these algorithms effective, have to be appropriately selected and designed. For object modeling, color, gradient, edge, texture, and motion are usually the features employed. However, several defects may appear when these features are used as the object model.

In this paper, a novel framework for video object segmentation and tracking for smart visual surveillance cameras is proposed with two key contributions. First, we propose a robust threshold decision algorithm for video object segmentation with a multibackground model. The proposed algorithm can determine a suitable best possible threshold value for our proposed multibackground model; hence, it can enable excellent performance for conditions with dynamic backgrounds without threshold tuning by users. In addition, it is based on a mechanism that is different from that of background subtraction-based video object segmentation, which can avoid possible error circulations. Second, we propose a video object tracking framework based on a particle filter with diffusion distance (DD) for measuring color histogram similarity and motion clues from video object segmentation. For enhanced tracking of nonrigid objects, we include color histogram in our object model as it is more stable for nonrigid moving objects.

2 Automatic Threshold Decision for Video Object Segmentation

The video object segmentation algorithm was established with a multibackground registration scheme to model complex and dynamic backgrounds. To make it fully automatic for alternative conditions, an automatic threshold decision technique that can automatically and accurately determine the threshold values for dynamic backgrounds is proposed.

![Fig.1. Block diagram for Video Object Segmentation](image)

2.1 Segmentation with Multibackground Registration

Our segmentation method is based on an online multilayer background modeling technique called multibackground registration (MBReg), whose block diagram is shown in Fig. 1. The key concept in this algorithm is the fact that it models the background with N layers of background images as an alternative of a single background layer. For each pixel position, the equivalent pixel in each layer of the background image represents one possible background pixel value. In Fig. 1, the background model is established and maintained in the MBReg and background update block. In the MBReg block, each input pixel of the current frame CurFrm(i, j, t), where (i, j) is the pixel position and t is the time index, is compared with the corresponding
background pixels in the multibackground image $BImg(i, j, t-1, k)$, where $k \in [1,N]$, and a matching flag, $\text{match}(i, j, k)$, is recorded by the following equation:

$$\text{match}(i, j, t, k) = \begin{cases} 1 & \text{if } BD(i, j, t, k) \leq BDth(i, j, t, k) \\ 0 & \text{otherwise} \end{cases}$$

(1)

in which $k \in \text{Built background layers}$, YUV color space is employed, $BD(i, j, t, k)$ denotes the background difference, which can be calculated using the following equations:

$$BD(i, j, t, k)y = |\text{CurFrm}(i, j, t)y - BImg(i, j, t-1, k)y|$$

(2)

$$BD(i, j, t, k)u = |\text{CurFrm}(i, j, t)u - BImg(i, j, t-1, k)u|$$

(3)

$$BD(i, j, t, k)v = |\text{CurFrm}(i, j, t)v - BImg(i, j, t-1, k)v|$$

(4)

and $BDth(\cdot)$ is the background threshold value for each color channel of each background pixel. In the background update block, an unmatched background counter, $\text{tSNO}(i, j, t, k)$, and a weighting coefficient, $Wgt(i, j, t, k)$, are attained to record the duration when a background pixel is unmatched to the input pixel and the confidence of each background pixel.

$$\text{CntSNO}(i, j, t, k) = \begin{cases} 0 & \text{if } \text{match}(i, j, t, k) = 1 \\ \text{CntSNO}(i, j, t-1, k) + 1 & \text{otherwise} \end{cases}$$

(5)

$$Wgt(i, j, t, k) = \begin{cases} Wgt(i, j, t-1, k) + 1 & \text{if } \text{match}(i, j, t, k) = 1 \\ Wgt(i, j, t-1, k) - 1 & \text{if } \text{CntSNO}(i, j, t, k) > \text{BDF}(i, j, t, k) \\ Wgt(i, j, t-1, k) & \text{otherwise} \end{cases}$$

(6)

where BDF is the background decomposing factor. With the unmatched background counter and weighting coefficient, the background model can be updated with the following equations:

$$BImg(i, j, t, k)y = \begin{cases} \text{UpdBckgnd}(i, j, t, k)y & \text{if } \text{match}(i, j, t, k) = 1 \\ 0 & \text{if } k \in \text{Built background layers} & \text{and } Wgt(i, j, t, k) < \text{RELth}(i, j, t, k) \\ 0 & \text{if } Wgt(i, j, t, k) = 0 \end{cases}$$

(7)

$$\text{UpdBckgnd}(i, j, t, k)y = \frac{\text{UpdBckgnd}(i, j, t-1, k)y \times Wgt(i, j, t, k) + \text{CurFrm}(i, j, t)y}{Wgt(i, j, t, k) + 1}$$

(8)

Built background layers $= \{k | Wgt(i, j, t, k) \geq \text{RegInth}(i, j, t, k)\}$

(9)

### 2.2 Proposed Threshold Decision

There are numerous threshold values in the proposed MBReg-based video object segmentation algorithm. They include $BDth(i, j, k, t)y$, $BDth(i, j, k, t)u$, $BDth(i, j, k, t)v$, $BDF(i, j, k, t)v$, $\text{RELth}(i, j, k, t)$, and $\text{RegInth}(i, j, k, t)$. In our experiential study, it was discovered that these thresholds can be set globally for all the positions and background layers without corrupting the quality of the generated object masks. That is, the indices $i, j,$ and $k$ can be removed from these thresholds. In addition, among these thresholds, $BDF(t)$, $\text{RELth}(t)$, and $\text{RegInth}(t)$ are used to control the update speed of the background model. The parameters are not sensitive in the proposed algorithm and can thus be set as constants or adjusted according to the requirements of different scenarios. However, the thresholds $BDthy(t)$, $BDthu(t)$, and $BDthv(t)$ used in (1) and (10) are important parameters that are highly correlated with the camera noise and should thus be temporally adjusted in order to address changes in lighting. In [15], a threshold decision method for the single background registration-based video segmentation algorithm was proposed. However, this algorithm cannot successfully and robustly determine the thresholds for conditions in which there are dynamic backgrounds. In this paper, we propose an
improved threshold decision algorithm for video object segmentation algorithm with multiple backgrounds.

![Block Diagram for Threshold Decision Algorithm](image)

**Fig. 2 Block Diagram for Threshold Decision Algorithm**

In addition to the ability to address dynamic backgrounds, the threshold decision algorithm required must have several wanted characteristics. First, it must be able to determine the optimal thresholds without any user input. Second, since background subtraction-based video object segmentation may cause an error circulation problem while updating the per-pixel background model, a threshold determination algorithm that is based on a different mechanism is required. Furthermore, the quantization effect of digital systems also has to be taken into concern. To meet these requirements, we proposed the threshold decision algorithm outlined in Fig. 2. The algorithm is based on the assumption that the camera noise is in the zero-mean Gaussian distribution, and the camera noise is the only factor affecting the optimal thresholds. The proposed algorithm consists of three sections: Gaussianity test, noise level estimation, and threshold decision.

1) Gaussianity Test: Before measuring the parameters of camera noise, the background of the input frame should be automatically indicated because it is very difficult to correctly measure the camera noise from the foreground. Since we assume that the camera noise is Gaussian distributed, the minimal background difference (BD\text{min}) of the background should also be Gaussian distributed:

\[
\text{BD}_{\text{min}}(i, j, t) = \text{Cur Frm}(i, j, t) - \text{Bl Frm}(i, j, k),
\]

where \(k\) is the frame number.

First, the frame is divided into a number of nonoverlapped \(M_b \times N_b\) blocks. The Gaussianity test is then applied to each block to determine if the minimal background differences in the block are Gaussian distributed or not. The Gaussianity test can be shown as the following equations:

\[
I_r = \frac{1}{M_b \times N_b} \sum_{m=1}^{M_b} \sum_{n=1}^{N_b} BD_{\text{min}}(m, n, r),
\]

\[
H(I_1, I_2, I_3, I_4) = I_3 I_4 I_1^2 - I_2^2 I_1^3 - I_2^2 I_1^3,
\]

where the smaller the \(H\) value, the closer the distribution of \(BD_{\text{min}}\) is to the Gaussian distribution, and \(Gth\) is the threshold value for binarizing the decision. If the minimal background differences in a block are Gaussian distributed, the block belongs to the background region because the (minimal) difference between the current frame and the background images is only caused by noise; that is, no foreground objects exist in this block to cause additional differences.

3 **Video Object Tracking**

**Particle Filter**
The objective of a particle filter is to estimate the posterior density of the state variables given the observation variables. The particle filter is designed for a hidden Markov Model, where the system consists of hidden and observable variables. The observable variables (observation process) are related to the hidden variables (state-process) by some functional form that is known. Similarly, the dynamical system describing the evolution of the state variables is also known probabilistically.

A generic particle filter estimates the subsequent distribution of the hidden states using the observation measurement process. Consider a state-space shown in Fig.3

![State Transition](image)

**Fig.3 Video Object Tracking with Hidden Markov Model**

The objective of the particle filter is to estimate the values of the hidden states $x$, given the values of the observation process $y$. The particle filter aims to estimate the sequence of hidden parameters, $x_k$ for $k = 0,1,2,3,...$, based only on the observed data $y_k$ for $k = 0,1,2,3,...$. All Bayesian estimates of $x_k$ follow from the posterior distribution $p(x_k | y_0,y_1,...,y_k)$. In contrast, the MCMC or importance sampling approach would model the full posterior $p(x_0,x_1,...,x_k | y_0,y_1,...,y_k)$.

The proposed object tracking algorithm with particle filter is shown in Fig.4. The objects obtained from the segmentation is given as input to particle filter, with the probability density function the objects which is moving to the next points are calculated and the tracking points are updated. The objects are initialized and released in object release block. The objects are tracked and their states are buffered in object state buffer with that multiple objects are tracked efficiently.

![Block Diagram](image)

**Fig.4 Block Diagram for Video Object Tracking**

4 Experimental Results

Our algorithm has been tested on indoor video sequences. We have tracked the person walking in the hall from the top view point. The proposed formulation has shown promising results. The system can segment the object and track the object even when the object is moving fast. The first set of figure illustrates the object segmentation with automatic threshold decision. Second set of figures shows the accurate tracking of the object with rectangle box. We show the tracking of the object with diffusion distance. Thus tracking of the object is made exactly with the proposed Algorithm. The following are the results of the proposed algorithm executed in MATLAB. Fig.5 shows the result of the proposed threshold decision algorithm. Fig.6 shows the result of proposed video object tracking algorithm. The proposed algorithm works well.
for tracking multiple objects in the video sequence. Fig.7 shows the results of tracking multiple objects in the outdoor. Thus algorithm works well for indoor and outdoor object tracking. The inputs are taken from the different available datasets and own dataset is also used to track objects in outdoor.
Fig. 6 Tracking of a human walking in Indoor
5 Conclusion

Video object segmentation and tracking is an attractive field due to numerous possible applications. We have proposed Automatic threshold decision algorithm for video object segmentation. We have used MultiBackground Registration scheme to update the background for various changes in lighting conditions. We have used Particle filter for tracking the object with Diffusion distance to compare color based model distribution in the HSV color space in order to strengthen the invariance to lighting condition. In our Particle filter, target modeling and tracking are achieved based on resampling around a predicted position obtained by the probability function. Our new formulation has been tested on datasets and the results on different sequences are shown. For future work, our approach can be easily extended to track multiple objects in the video.

References


