RBF Approach to Background Modelling for Background Subtraction in Video Objects

Sivabalakrishnan.M,
Research scholar, Department of CSE,
College of Engineering, Anna University
Chennai, India

Dr.D.Manjula
Assistant Professor, Department of CSE
College of Engineering, Anna University
Chennai, India

***Abstract*: Background Subtraction is a widely used approach to detect moving objects from static cameras. Many different methods have been proposed over the recent years and there are a number of object extraction algorithms proposed in the literature, most approaches work efficiently only in constrained environments where the background is relatively simple and static. Nowadays background modeling and subtraction algorithms are commonly used in object detection and tracking applications. The goal is to obtain a clean background and then detect moving objects by comparing it with the current frame. This paper presents a RBF neural network architecture is proposed to form an unsupervised Bayesian classifier for this application domain. The constructed classifier efficiently handles the segmentation in natural-scene sequences with complex background motion and changes in illumination. The weights of the proposed RBF serve as a model of the background and are temporally updated to reflect the observed statistics of background. The segmentation performance of the proposed RBF neural network is qualitatively and quantitatively examined and compared with fuzzy and classical background subtraction method. Our experimental results demonstrate that proposed system is much more efficient, robust and accurate than classical approaches**

Key words: Background Subtraction, Foreground Detection, neural network, RBF

I. .INTRODUCTION

Background extraction is an important part of moving object detection algorithms that are very useful in surveillance systems Background subtraction is commonly used in the field of video surveillance, optical motion capture, and multimedia application where it needs in the first step to detect the moving objects in the scene. The basic idea is to classified pixel as background or foreground by thresholding the difference between the background image $Bt(x, y, t)$ and the current image $It+1(x, y, t)$. According to importance of real-time computations in the surveillance systems, subtraction methods are so significant. Due of the presence of critical situations, false positive or negative detection appear corresponding to false classification of pixels

Due of the presence of critical Situations. The detection and segmentation of moving objects in natural video sequences is an important requirement for multimedia indexing and retrieval, but also for adaptation. With the advent of broadly available mobile video players, it has become desirable to adapt video contents to small screens by choosing a suitable compromise between scaling and cropping [1]. This however requires the detection of video-objects that might be of interest. Approaches like Kopf's automatic scaling and cropping rely on pre-trained detectors for fixed kinds of objects like faces or superimposed text [2]. In contrast, background subtraction only relies on the assumption that an objects moves. In our generalized approach, it is only necessary that objects move different from global motion. In surveillance applications, background subtraction has become a standard method for video object segmentation. Even recent publications like [2] still use the scheme that originates from the publication by Stauffer and Grimson [7] who proposed to model the background image pixel-wise by Gaussian Mixture Models (GMM). This exploits the fact that background pixels in subsequent frames of a video should be highly correlated and therefore can be described by average color and variance. See [1] for an introductory overview. For a non-static camera however, the correspondences between background pixels are not given by their fixed coordinates but have to be estimated with a global motion model. The estimation of global motion as well as choosing a suitable model representation are still unsolved problems. For background subtraction however, pixel exact mosaics are required and it has been stated in literature that it is "impractical or even impossible to use a single background image" for a whole shot .

Our approach is to extract the object from the background and fill the corresponding image region with white pixels so as to block the identifying features. Our approach is to extract feature information from the object and develop statistical models, such as Hidden Markov Models, to model and track. In this paper, we address those problems through the use of fuzzy logic.The rest of this paper is organized as follows. In section II, overview of Background subtraction is illustrated. In, section III show the propose scheme and RBF architecture are presented in detail. Section IV shows the experimental results. Finally the conclusion is given in section V.

II OVERVIEW OF BACKGROUND SUBTRACTION

Background Subtraction, which generates a foreground mask for every frame. This step is simply performed by subtracting the background image from the current frame. When the background view excluding the foreground objects is available, it becomes obvious that the foreground objects can be obtained by comparing the background image with the current video frame. By applying this approach to each frame one effectively achieves tracking any moving object a background image can be elegantly used to determine the foreground objects by comparing the input frame with the background image and mark the differences as foreground objects. This technique is commonly known as background subtraction or change detection. It is the most popular approach in video surveillance applications, because it is a computationally efficient technique and it is relatively easy to obtain background images for static surveillance cameras. In practice, camera noise and regions in which the object has the same color as the background make the separation of foreground objects and background more difficult. Finally, a few post processing filters are presented that can remove obvious errors like small clutter regions.

We assume that the camera is fixed. We formulate object extraction as an adaptive classification problem. The input video frame is partitioned into small blocks, for example, 4x4 blocks. For each block, a classification decision is made: the block belongs to the changes or not. To this end, we extract invariant features from the image blocks, as illustrated in Fig. 1. Based on these features, we build a statistical model for the background and classify the image blocks into two categories: foreground and background. Because the background is time-varying, the background model and the classifier should be adaptive. However, background adaptive is also risky. For example, if a person sits still or sleeps in the couch for a long time, say hours, the adaptive background model will consider the person part of the background. This is not acceptable because of unprotected privacy. To solve this problem, we fuse the high-level knowledge obtained from object tracking with the low-level feature-based classification so as to guide the background update. At this stage, the foreground may still contain objects. To address this issue, we develop a decision process based on a fuzzy logic inference system to detach these objects from the background. In the following sections, we explain the proposed background subtraction scheme in detail.

III .PROPOSED METHOD

The basic idea that forms the basis of all probabilistic background modelling and video objects segmentation approaches discussed in Section II and the one presented here is a direct consequence of the definition of the background stated in the introduction, Feature values corresponding to background objects will occur more often than those pertinent to the foreground. In addition to this assumption, these methods share a set of common tasks that need to be performed to learn, update, and store the background model that enables efficient segmentation [3], [4]. These tasks, which have been used as guidelines in the design of BNN, are as follows

1) Storing the values of the pixel features and learning the probability with which each value corresponds to background/ foreground;

2) Determining the state in which new feature values should be introduced into the model (i.e., when the statistics already learned are insufficient to make a decision);

3) Determining which stored feature value should be replaced with the new value.

The two latter requirements are consequences of the fact that real systems are limited in terms of the number of feature values that can be stored to achieve efficient performance. In terms of the NN implementation proposed here, this translates into the number of patterns stored, i.e., the number of neurons used per pixel.

An RBF network is a three-layer feed forward neural network which consists of an input layer, a hidden layer and an output layer.

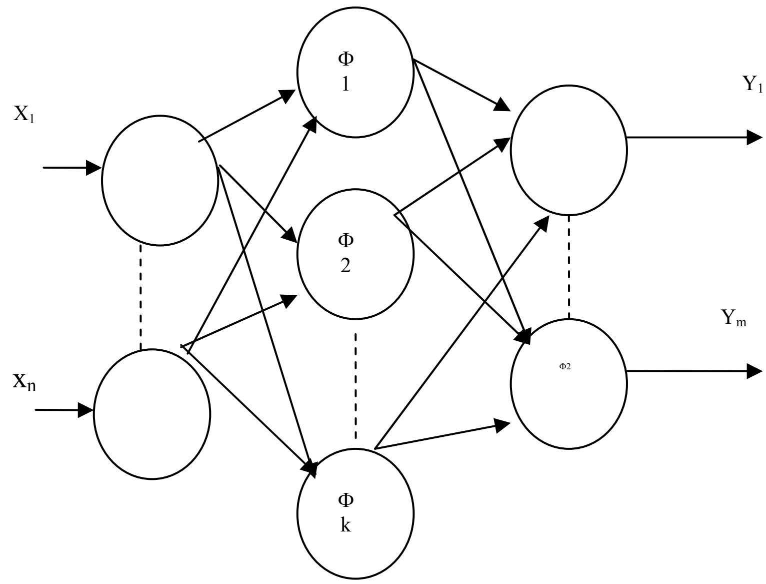

Figure 1. RBF network Structure

The structure of a traditional RBF network is shown in figure 1. The number of neurons in input layer depends on number of input attributes of training samples. The number of neurons in the output layer depends on number of target class labels. Assume that *x* is an *n*-dimensional input vector, there are *m* neurons in output layer k neurons in hidden layer. It can be formulated as

$$\varphi_j(x) = e^{-\ \| \xi - \chi\varphi\| / 2\sigma\varphi 2} \tag{1}$$

$$y_i = \sum_{j=1}^{k} W_{ij}\phi_j(x) \tag{2}$$

where $i=\{1,2,\ldots,m\}$, $j=\{1,2,\ldots,k\}$, $\Phi_j(x)$ is the output of the *j*th hidden node, C_j is the center vector of Gaussian kernel function which has the same number of dimension with *x*, and σ_j is the width of Gaussian kernel function at the *j*th hidden node. Besides, y_i is the output of the *i*th output neurons and W_{ij} is the connect weight from the *j*th hidden node to the *i*th output node.

As neural networks have the ability to classify input vectors into different classes, they can be used in classifying regions of video frames into background or foreground classes. Another property of neural networks is their ability to estimate highly non-linear functions. They can also represent a many-to-one mapping. These properties of neural networks are enough to convince one to use them in case of video object classification; i.e. foreground background segmentation.

In order to segment a video frame into foreground and background regions first we divide it into small blocks. By dividing an image, here a video frame, into blocks we can extract features form the blocks and decide whether a block is a background block or not. For each block of the video frame we make different neural networks and these networks are trained separately. In training stage each neural network will be trained by samples of background in the position of its corresponding block. There are several types of Neural Networks that can be used directly or after post processing the results . In our application we used RBF networks.

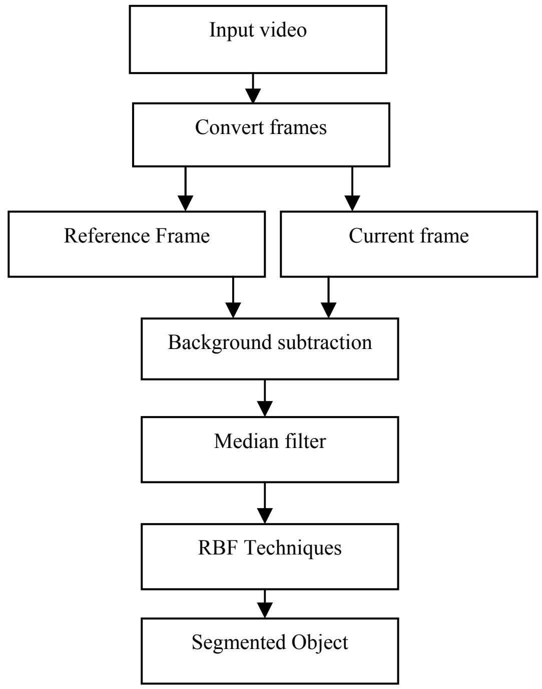

Figure 2. The overall architecture of proposed system.

A. Learning Procedure

Most of the existing learning algorithms for RBF neural network employ different schemes for updating the output weights, i.e., the weights that connect the RBF with the output units, and the centers of the RBF, i.e., the vectors in the input space that represent the prototypes of the input vectors included in the training set [14]. In this section, the incremental learning method was used to construct a classification model because of its simple and easy implementation. We attempted to train the network to implement a desired input-output mapping by producing incremental changes of the weights of the network. If the responses of the radial basis functions are not substantially affected by incremental changes of the model, then the learning process reduces to incremental changes of the weights of the associative network which alone are unlikely to implement non-trivial input-output mappings. The ability of the network to implement a desired input-output mapping depends very strongly on the sensitivity of the responses of the radial basis functions to incremental changes of their corresponding model.

B. Training Procedure

The RBF networks employ a hybrid two-stage training scheme which decouples the learning task from both hidden and output layers and thus eliminates the need for the slow back error propagation. The sum of the squared error SSE criteria is consider as the error function, and is given below.

$$E = \sum_{=1}^{r} (O_k^D - O_k^R)^2 \qquad (3)$$

Where O_k^D is the desired output at the *k*th node commuted from any analytical approach and O_k^R is the RBF network output at the *k*th node computed for the test pattern, and *r* the number of output neurons. In the training process, the error function *E* is minimized over the given training set by adaptively updating the free parameters of the RBF network. These parameters are the RBF centers(μ_j, *s*), their widths (σ_j,*s*) and the second layer weight (w_{kj} *s*). The RBF network is trained in following three steps:

Step—1

Determine the hidden node (RBF) centers.

Step—2

Determine the hidden node (RBF) widths.

Step—3

Determine the second layer connection weights

Several methods have been reported in the literature to find the proper hidden node centers. A commonly used method is to choose arbitrarily some data points from the input domain as centers. However, such a method can't guarantee adequate performance as it may not satisfy the requirement that centers should suitably sample the input domain. In order to achieve a given performance, an unnecessarily large RBF network may be required which causes computation complexity and often numerically ill-conditioning. On the other hand, the orthogonal least square (OLS) method choosing RBF centers one by one is a rational way until an adequate network has been constructed, and therefore is used in the present work. After the hidden node centers are found, their widths can be determined by one of the several heuristics to obtain a close enough approximation to the desired output. It has been suggested in the literature that the RBF networks with a single global fixed value 'σ' for all 'σ_j' values have the capability of universal approximation. In order to preserve the local repose characteristics of the hidden units, a relatively small (positive) value for this global width parameter should be chosen. However the widths of all RBF units are taken to be equal which is known as the spread factor (SF) of the RBF network. If SF is too small or too large,

IV EXPERIMENTAL RESULT

The following experimental results on several sequences show that the proposed RBF NN system for moving object detachment not only preserves the advantage of the low-level feature-based object extraction algorithm. These results also verify the robustness of the proposed detaching algorithm. The fuzzy system even has the added benefit of removing some of the dark shadows that our color-based algorithm fails to detach. However, some of the hardened silhouettes appear slightly eroded. Since the goal is to perform activity analysis, the eroded object should not cause serious problems, although we are investigating better algorithms to fill out the fuzzy shape. While the RBF NN system performs better than the corresponding crisp one, a good portion of the adaptive background is lost. As the sequence progresses, this error is compounded. Hence, we need more research on techniques to "rebuild" the displayed crisp silhouette from the fuzzy silhouette and methods to reacquire a good silhouette occasionally during the sequence. Since we will process long sequences of activity, a reasoning module should be able to detect when the object is stationary for a short time. Then, we believe that we can "reset" the golden standard silhouette.

a) Quantitative Evaluation

Each pixel in a background subtraction method's classification was determined to be: true positive for a correctly classified foreground pixel, false positive for a background pixel that was incorrectly classified as foreground, true negative for a correctly classified background pixel, and false negative for a foreground pixel that was incorrectly classified as background. the different methods can be evaluated by the calculation of T_P, T_N, F_P, F_N, where T_P, T_N, F_P, F_N means the number of true positive, true negative, false positive and false negative, respectively. where T_P is the number of background pixels found as background, T_N is the number of foreground pixels found as foreground , F_P is the number of background pixels images found as foreground (false positives) and F_N is the number of foreground pixels found as background. images found as normal (false negatives). Sensitivity and specificity are also referred to as the true positive rate (TPR) and true negative rate (TNR), respectively. After every pixel had been classified into one of those four groups, the sensitivity, the specificity was calculated.

Sensitivity and specificity are statistical measures of the performance of a binary classification test.**.** Sensitivity (also called recall rate in some fields) measures the proportion of actual positives which are correctly identified .Specificity measures the proportion of negatives which are correctly identified. These two measures are closely related to the concepts of type I and type II errors. sensitivity is defined in equation 4, specificity is defined in equation 5

$$\textit{Sensitivity} = TP / (TP + FN) \quad (4)$$

$$\textit{Specificity} = TN / (FP + TN) \quad (5)$$

The sensitivity measures the proportion of actual positives which are correctly identified. The specificity measures the proportion of negatives which are correctly identified. Table 1 shows the quantitative results of various frames.

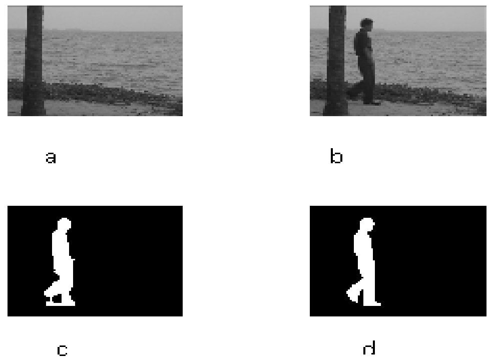

Figure 3. (a) background reference image (b)a frame from video (c) result of proposed method (d) ground truth image

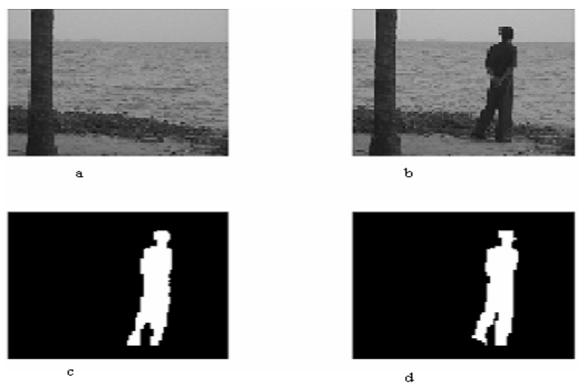

Figure 4. (a) background reference image (b)a frame from video(c) result of proposed method (d) ground truth image

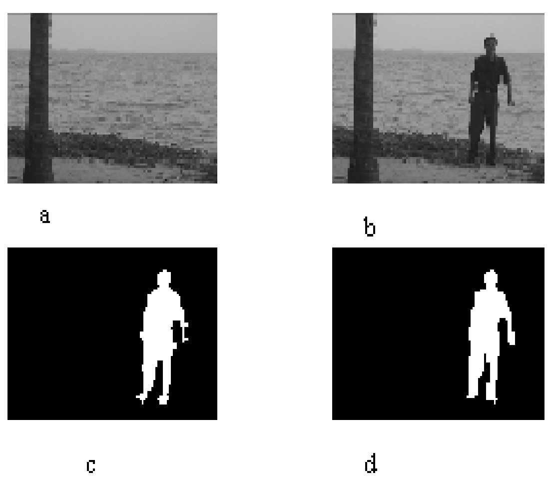

Figure 5. (a) background reference image (b)a frame from video(c) result of proposed method (d) ground truth image

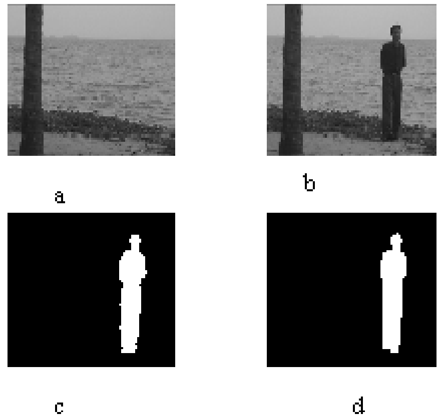

Figure 6. (a) background reference image (b)a frame from video(c) result of proposed method (d) ground truth image

TABLE I. QUANTITATIVE EVALUATION AND COMPARISON OF THE TEST RESULTS

Sample	Method	Tp	Tn	Fp	Fn	Sensitivity	Specificity
1	BGS	1482	18880	136	136	0.9159	0.9928
	FBGS	1466	18880	120	120	0.9243	0.9937
	NBGS	1466	18880	119	119	0.9561	0.9982
2	BGS	1716	18651	100	100	0.9449	0.9947
	FBGS	1700	18651	83	83	0.9534	0.9956
	NBGS	1700	18651	81	81	0.9706	0.9932
3	BGS	1832	18548	93	93	0.9517	0.9950
	FBGS	1829	18548	86	86	0.9551	0.9954
	NBGS	1828	18548	82	82	0.9661	0.9974
4	BGS	1689	18741	43	43	0.9752	0.9977
	FBGS	1687	18741	38	38	0.9780	0.9980
	NBGS	1685	18741	35	35	0.9796	0.9986

V. CONCLUSIONS

In this paper, we proposed a accurate and robust object extraction scheme for a dynamic environment. A novel background modelling and subtraction approach for video object segmentation in complex sequences has been proposed. The proposed method is probabilistic and relies on an RBF Neural Network to achieve estimation of required pdfs and segmentation. The approach was evaluated on a set of diverse sequences, pertinent to the automatic surveillance application domain. Good segmentation results have been obtained for these complex sequences. The proposed approach represents an improvement in segmentation ability when compared to a well-known Fuzzy and classical background subtraction method. This result indicates that the proposed approach could benefit from introduction of adaptive learning

The approach would also benefit from the introduction of mechanisms that would allow it to exploit spatial information, typically used in still-image segmentation. Currently, the extension of the approach to use the feedback from higher processing modules of object tracking to enhance the segmentation is being examined. To deal with the challenges of object extraction in dynamic environment, we train high-level knowledge and low-level features and developed a RBF neural network system. The results on several sequences show that this algorithm is efficient and robust for the dynamic environment with new objects in it. To compare background subtraction with their RBF Neural Network approaches in real world both classical and fuzzy algorithms .Experimental result shows that neural approach is more accurate than classic and fuzzy approach. We are currently working on making the prediction more accurate and creating a scheme to recover missing moving parts using known results from feature-based classification. Also, we intend to study the impact of the accuracy of results on the performance of future activity modeling and analysis

REFERENCES

[1] A. Elgammal, D. Harwood, L. Davis, "Non-parametric Model for Background Subtraction", *6th European Conference on Computer Vision*. Dublin, Ireland,June/July 2000

[2] M.H Sigari, "Fuzzy Running Average and Fuzzy Background Subtraction: Concepts and Application" , IJCSNS International Journal of Computer Science and Network Security , Vol.8, No.2, February 2008.

[3] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans.Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[4] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002

[5] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "Neural network approach to background modeling for video object segmentation,"*IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1614–1627, Dec.2007

[6] T. Horprasert, D. Harwood, L. Davis, "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection", Proc. IEEE ICCV'99 FRAMERATE Workshop, Kerkyra, Greece, September 1999.

[7] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Computer Vision* and Pattern Recognition, 1999, pp. 246–252.

[8] George J. Klir, Bo Yuan, "Fuzzy Sets and Fuzzy Logic", Prentice Hall, 1995."Fuzzy Logic Toolbox User's Guide", The Math Works, Inc. 1998