1 234

Neural Network Approach to Background Modeling for Video Object Segmentation

Dubravko Ćulibrk[+], Oge Marques[*], Daniel Socek[†], Hari Kalva[♯], Borko Furht[♭]

***Abstract*— The paper presents a novel background modeling and subtraction approach for video object segmentation. A 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 neural network serve as a model of the background and are temporally updated to reflect the observed statistics of background. The segmentation performance of the proposed neural network is qualitatively and quantitatively examined and compared to two extant probabilistic object segmentation algorithms, based on a previously published test pool containing diverse surveillance-related sequences. The proposed algorithm is parallelized on a sub-pixel level and designed to enable efficient hardware implementation.**

***Index Terms*— Object segmentation, Neural networks, Video processing, Background subtraction, Automated surveillance.**

I. Introduction

THE rapid increase in the amount of multimedia content produced by our society is a powerful driving force behind the significant scientific effort spent on developing automatic methods to infer meaning of this content. A vital part of this work is directed towards the analysis of video sequences. Object segmentation represents a basic task in video processing and the foundation of scene understanding, various surveillance applications, as well as the emerging research into 2D-to-pseudo-3D video conversion. The task is complex and is exacerbated by the increasing resolution of video sequences, stemming from continuing advances in the video capture and transmission technology. As a result, research into more efficient algorithms for real-time object segmentation continues unabated.

In this paper, a common simplifying assumption that the video is grabbed from a stationary camera is made. The task is still difficult when the segmentation is to be done for natural scenes where the background contains shadows and moving objects, and undergoes illumination changes. In this context, the basic segmentation entities can be defined as follows:

- *All objects that are present in the scene, during the whole sequence or longer than a predefined period of time, are considered background objects.*
- *All other objects appearing in the scene are referred to as foreground.*

[+]dculibrk@fau.edu, [*]oge@cse.fau.edu, [†]dsocek@fau.edu, [♯]hari@cse.fau.edu, [♭]borko@cse.fau.edu

This work was supported by the Center for Coastline Security and Center for Cryptography and Information Security.

Authors are with Florida Atlantic Univeristy.

Manuscript received xxxx xx, 2006; revised xxxx xx, 2006.

The goal of the video object segmentation is to separate pixels corresponding to foreground from those corresponding to background.

If the state of the background is known for every frame of the sequence and there are no changes in illumination, the segmentation can be accomplished by a simple comparison between the background image and a frame of the sequence. This, however, is unrealistic for almost all applications. In the absence of an exact model for the background, one has to be estimated based on the information in the sequence and some assumptions. The process of modeling the background and determining the foreground by comparison with the frames of the sequence is often referred to as *background subtraction*.

Two broad classes of background-subtraction methods can be identified:

1) Filter based background subtraction.
2) Probabilistic background subtraction.

Filter based approaches were developed first and rely on some sort of low-pass filtering of the frames of the sequence to obtain a model of the background in the form of a background image. Their main weakness is the inherent assumption of the background changing more slowly than the foreground. High frequency motion in the background such as that of moving branches or waves often leads to misclassification of these background objects. This makes filter based unsuitable for applications with complex and dynamic background changes [1] [2] [3]. They are computationally inexpensive when compared to probabilistic methods, but are unable to achieve good segmentation results for many natural scenes.

Probabilistic methods are an effort to escape the limitations of the filter-based approaches by learning the statistics of the pixels corresponding to background and using them to distinguish between the background and the foreground. They are the preferred approach for segmentation of sequences with complex background. Their main shortcoming is that they are computationally complex and only able to achieve real-time processing of comparatively small video formats (e.g. 120×160 pixels) at reduced frame rates (e.g. 15 frames per second) [4].

The development of a parallelized probabilistic object-segmentation approach, which would allow for efficient hardware implementation and object detection in real-time for high-complexity video sequences (in terms of the frame size as well as background changes), is the focus of this paper. In this respect it is an extension of previously published work [5] [6] pertinent to marine surveillance. Here, the proposed approach is described in more detail and examined in terms of applicability to a broader range of surveillance application.

With this in mind, it is tested on a diverse set of surveillance related sequences compiled by Li *et al.* [4].

The proposed solution employs a feed-forward neural network to achieve background subtraction. To this end, a new neural network structure is designed, representing a combination of Probabilistic Neural Network (PNN) [7] [8] and a Winner Take All (WTA) neural network [9]. In addition, rules for temporal adaptation of the weights of the network are set based on a Bayesian formulation of the segmentation problem. Such a network is able to serve both as an adaptive model of the background in a video sequence and a Bayesian classifier of pixels as background or foreground. Neural networks posses intrinsic parallelism which can be exploited in a suitable hardware implementation to achieve fast segmentation of foreground objects.

Section II provides insight into our motivation and a survey of related published work. Section III holds the discussion of the Bayesian inference framework used. Section IV describes the main aspects of the proposed approach. Section V is dedicated to the presentation and discussion of simulation results. Section VI contains the conclusions.

II. Motivation and Related Work

The segmentation approach described here is motivated by previous work in the domain of marine surveillance [5] [6]. It is intended as a solution to the problem of segmentation in natural-scene, complex-background sequences. The ultimate goal of the project is to achieve real-time segmentation of high-resolution QuadHDTV images (frame size of 3840×2160 pixels). The segmentation module should eventually be implemented as a hardware component embedded in a QuadHDTV camera; hence our interest in a hardware-friendly solution.

The discussion of published works in this section is localized to probabilistic background-subtraction approaches to foreground segmentation, with the exception of three papers concerning the application of neural networks to computer vision problems [10] [11] [12]. The motive for the former is the comparison of the proposed approach with other probabilistic methods used to address the same problem. The latter are included to point out the differences between the previous relevant applications of neural networks and the approach proposed.

The Kalman filter approach to video object segmentation [13] can be considered one of the first probabilistic approaches applied. It is an optimal solution to Bayesian estimation when the changes in the background are modeled by a dynamic linear process with normal distribution of errors in the measurement of the pixel values used to determine the state of the system. In the work referenced, the state of the system has been defined as the vector of the change in gray values of the pixel and its first derivative. The linearity assumption proved to be too restrictive to enable efficient segmentation in the case of complex background scenes with high-frequency changes. Nevertheless, extended Kalman filter remains the only probabilistic approach that considers the error in the measurements explicitly.

While the Kalman filter approach makes an assumption about the normal (Gaussian) distribution of the noise, other early probabilistic approaches assumed normal distribution of the values of a single pixel. Thus, they tried to approximate the Probability Density Function (PDF) of these values by a single Gaussian, whose parameters are recursively updated in order to follow gradual background changes within the video sequence [14]. These techniques achieve slightly better segmentation results than the Kalman filter.

A natural extension to the single Gaussian-based approaches are the Mixture of Gaussians (MoG) models. These methods use multiple evolving Gaussian distributions as a model for the values of the background pixels [15] [3] [16]. They showed good foreground object segmentation results for many outdoor sequences. However, weaker results were reported [17] for video sequences containing non-periodical background changes. The improved performance of these methods can be attributed to the fact that they do not incorporate the assumption of the normal distribution of background pixel values. The shape of the PDFs they are trying to estimate can be any shape that can be approximated with a predetermined number of Gaussian curves. In fact, with an infinite number of Gaussian curves one can approximate any curve [18]. In that case, the Gaussian curves represent what will be referred as a kernel in the terminology of the approach proposed below. For reasons of computational complexity, the typical number of Gaussians used for modelling is 3-5 [3]. This yields a rather inaccurate approximation of the PDFs and is the reason behind the poor performance in complex sequences.

Recently (2003), Li et al. proposed a method for foreground object detection, which represents a combination of a filtering and a probabilistic approach [4]. It can therefore be considered a hybrid in terms of the classification above. Initial segmentation is filter-based and the model maintains a reference background image as a model of the background. The authors propose a novel approach to cope with the inability of the filter-based approaches to differentiate between the movements of the foreground objects and background objects in complex scenes. They model the PDFs of the pixel values detected by the initial segmentation. Thus, they are able to distinguish between the errors in the initial segmentation and the true foreground pixels. The PDFs are updated in time and a Bayes' rule based decision framework is formulated based on the assumption that the pixel values observed more often at a single pixel are more likely to be due to background object movement. The applicability of the stated assumption to the data that has been filtered through the initial segmentation is unclear. Nevertheless, the approach is able to achieve good segmentation results for sequences containing high-frequency changes in the pixels pertinent to background.

PNNs have not, to the best of our knowledge, been applied to achieve motion-based object segmentation. They have been used to enhance the performance of certain specific object-segmentation methods as reported in [11]. The PNN was used to enhance the segmentation results of a color-based classifier used to detect humans in specific scenes. The approach used a model foreground objects, rather than background. However, a supervising classifier was used to generate the training set for a PNN and to periodically retrain it, differing from the fully unsupervised approach proposed here.

An interesting application of PNNs in the domain of computer vision is reported in [10]. The PNN was used for cloud classification and tracking in satellite imagery. The PNN was again a supervised classifier, and the approach did not incorporate background modeling.

Both applications of neural networks to computer vision problems, discussed above, are characterized by the use of certain problem specific classifiers to supervise the neural network. In addition, the training of the network is not incremental and both approaches require the network to be retrained periodically.

More recently (2006), Pajares [12] proposed a Hopfield Neural Network (HNN) based algorithm for change detection. As it is the case with the two preceding algorithms, this algorithm does not employ background modeling to achieve segmentation segmentation. The approach could potentially be used as a part of a filter-based approach instead of more traditional frame difference calculation methods. This would, however, severely increase the computational requirements, which represent a major advantage of filter-based approaches.

III. Probabilistic Pixel Classification and Background Model

The goal of the probabilistic segmentation algorithm presented is to be able to classify the pixels in a frame of the sequence as foreground or background, based on statistics learned from the already observed frames of the video sequence. Different pixel features, such as intensity or RGB components, can be used as basis for segmentation. The value of these features changes with each new frame of the sequence. If the pixel feature used for segmentation is intensity, and the numerical intensity value of a pixel at frame t of the sequence is for example 152, than the pixel feature value for that pixel at frame t corresponds to its intensity value (152). In fact, the video sequence itself can be viewed as a set of pixel feature values varying over time. Stauffer *et al.* [3] refer to these changing pixel feature values as "pixel processes". More formally: if $l = (x, y)$ is the location of a single pixel within the frames of the video sequence, then a pixel process of pixel l is a set of all feature values of the pixel for all the frames in the sequence:

$$PP_l = v_l^t : t \in 0, ..., T \tag{1}$$

where PP_l is the pixel process at pixel l, T is the number of frames in the sequence, and v_l^t is the feature value of the pixel l at frame t.

The observed features of pixels can be scalar in nature such as intensity or vectors (e.g. RGB values). Also, the features used for classification can be some higher-level features extracted for location l. All that is required is that in a certain frame t the algorithm is able to decide whether the pixel at arbitrary location l is pertinent to background or foreground, given the values v_l^t and part of the process PP_l up to current frame t_c. In the subsequent discussion, whatever the nature of the features actually used for classification, their value for a certain pixel will be simply be referred to as pixel value.

Fig. 1 shows a plot of two sample pixel processes containing some 300 pixel values, corresponding to 10 seconds (300 frames) of a sample sequence. Plot 1(b) corresponds to a pixel of the water-surface in the frame shown in 1(a), while 1(c) is a pixel pertinent to section of fig3 on the far right of the frame.

In the proposed algorithm, the background model serves as the exclusive repository of the statistical information extracted form the observed parts of pixel processes. To classify pixels, a strategy that minimizes the expected risk of our decision is employed. Such strategies are known as Bayesian [19].

A. Bayes Classification Strategy

The segmentation problem is formulated to enable the use of Bayes decision rule to achieve segmentation. For a certain frame t, we are trying to estimate the dependent variable ($\Theta_l \in \{f, b\}$). The event of pixel at location l being part of the foreground corresponds to $\Theta_l = f$, while $\Theta_l = b$ when the pixel is pertinent to background. Θ_l is a function of the random variable V taking values in the space of pixel feature values. Note that Θ_l is itself a random variable. Using a Parzen estimator we can construct the estimate of the PDF $p_l(V)$ of V. Although no direct information of the distribution of Θ_l is available, suppose that one is able to infer some other knowledge about the conditional probability distribution $p_l(\Theta_l|V)$ of values of theta occurring when a certain value of V has been observed. Then, a Bayesian decision rule that allows for the classification of pixels is formed as follows:

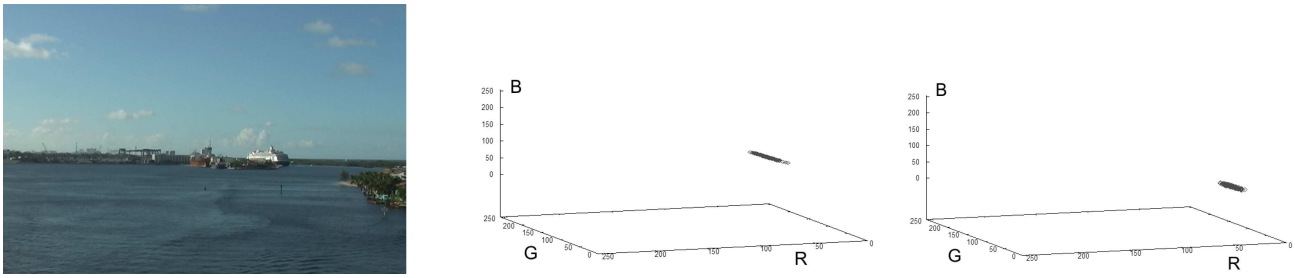
$$\Theta_l = \begin{cases} f, & c_b p_l(f|v)p_l(v) > c_f p_l(b|v)p_l(v); \\ b, & c_b p_l(f|v)p_l(v) \leq c_f p_l(b|v)p_l(v). \end{cases} \tag{2}$$

The costs are application-dependent and determined subjectively. In experiments presented in this paper they are considered to be the same (i.e. $c_f = c_b$). This means that the misclassification of a pixel is considered equally bad if it is labelled as foreground or as background. Thus, only knowledge of the PDF ($p_l(V)$) and the prior conditional probabilities of background and foreground occurring at pixel l is needed to classify the pixel. However, the PDF $p_l(V)$ and its shape is unknown, as is the case with the prior probabilities $p_l(f|v)$ and $p_l(b|v)$, too. To classify the pixels they have to be estimated. This must be done efficiently if one hopes to achieve real-time performance and segmentation of large frames.

B. Background Model

At each given frame (t) of the sequence, the background model stores the values of estimated probabilities ($p_l(V)$, $p_l(f|v)$ and $p_l(b|v)$) for each pixel (l) of the frame. The prior probabilities $p_l(f|v)$ and $p_l(b|v)$, for a specific value v are scalar values and can be stored efficiently. The values of the PDF $p(V)$ should, in general, be known for any pixel value v. This makes storing of the estimated PDF a significant problem.

If a certain shape is assumed for the PDF, it can be efficiently represented by the parameters of the proposed distribution [14] [3]. When this is not the case, the naïve approach is to store the complete histogram of the PDF. Assuming that the feature used for classification is the RGB value of the pixels, coded as 24 bits, this would result in a structure containing 256^3 (≈ 16.8 million) entries per pixel.



(a) Sequence frame. (b) Water-surface pixel. (c) Tree pixel.

Fig. 1. Sample pixel process plots.

However, as Li *et al.* [4] show, it may not be necessary to store the whole histogram. They make a case for the assumption that the features that are part of the background tend to be located in the small subset of the histogram, since the processes occurring in the background tend to be repetitive. Plots in Fig. 1 illustrate this effect, since the RGB values concentrate in small parts of the entire space of feature values. They were able to achieve adequate segmentation results for complex-background sequences by storing the statistics for 80 values (they covered $4.77ppm$ of the histogram) [17]. For other features that spanned an even larger space, Li *et al.* performed binning of the features, considering all values to be the same if they differed by less than 3 in each dimension.

In previously published work [6], color-based segmentation was employed. RGB values were used as features for pixel classification. Each channel was coded with 8 bits. Here, we turn to intensity of the pixel as a low-level feature, to perform base-line evaluation of the approach. The intensity is calculated as an average of the RGB values and coded with 8 bits. Instead of representing the PDF in the form of histogram and applying the binning procedure, the PDF is estimated using Parzen estimators [18]. A Parzen estimator of a PDF based on a set of measurements has the following analytical form:

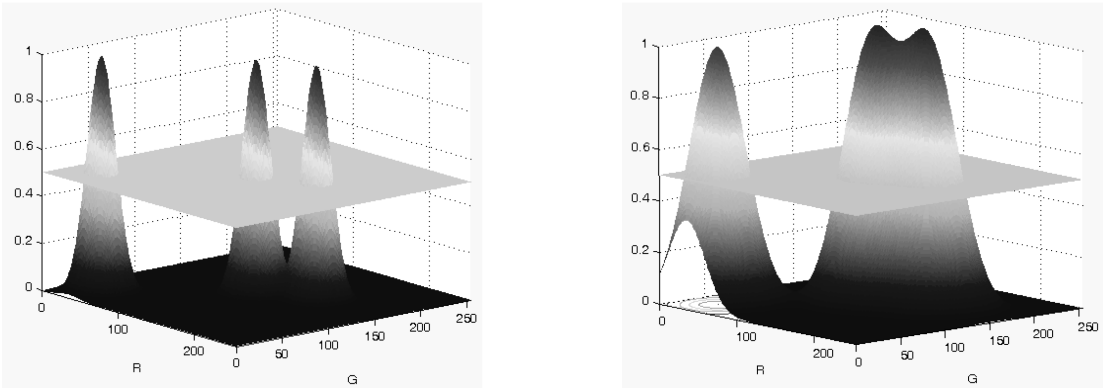
$$p(v) = \frac{1}{(2\pi)^{n/2}\sigma^n}\frac{1}{T_o}\sum_{t=0}^{T_o}\exp[-\frac{(v-v_t)^T(v-v_t)}{2\sigma^2}] \quad (3)$$

where n is the dimension of the feature vector, T_o is the number of patterns used to estimate the PDF (observed pixel values), v_t are the pixel values observed up to the frame T_o, σ is a smoothing parameter.

The Parzen estimator defined by (3) is a sum of multivariate Gaussian distributions centered at the observed pixel values. As the number of observed values approaches infinity, the Parzen estimator converges to its underlying parent density, provided that it is smooth and continuous. The smoothing parameter controls the width of the Gaussians and its impact on the representation is treated in more detail below. The scaling factor preceding the sum in (3) has no impact when Bayes decision rule (2) is used, and can be discarded. Using a Parzen estimator-based approach it would be enough to store all the values of a certain pixel observed in the known part of the sequence. This would still be inefficient, especially since the values of the background pixels concentrate in a small part of the value space and are therefore similar. A representation based on a relatively small number of Gaussians can be achieved if each Gaussian is used to represent a portion of observed patterns which are similar according to the predefined threshold [8]. The procedure is similar to the binning used by Li *et al.*, but the resultant PDF representation retains the notion of how close a value that is assigned to a particular Gaussian is to its center. This is not the case with the binning, since all the values within a bin are assigned the same value of PDF. Fig. 2 shows the plot of a Parzen estimator for three stored points with values in two-dimensional plane (e.g. if only R and G values for a pixel are considered). The horizontal planes in Fig. 2 represent the threshold (θ) values used to decide which feature values are covered by a single Gaussian. All features within the circle defined by the cross-section of the Parzen estimate and the threshold plane are deemed close enough to the center of the peak to be within the cluster pertinent to the Gaussian. The selection of smoothing parameter value and the threshold controls the size of the circle and the cluster. Larger values of σ lead to less pronounced peaks in the estimation, i.e. make the estimation "smoother". For a fixed smoothing parameter value, lower values for the threshold θ will lead to larger coverage of the space of feature values by the estimate.

IV. Background Modeling Neural Network (BNN)

In 1990, Specht [7] introduced a neural network architecture to be used as a Bayesian classifier, based on the Parzen estimation of the PDFs involved, and a Bayes decision rule given by (2). He dubbed these networks Probabilistic Neural Networks (PNNs). This architecture is a natural way to implement the classifier described in Section III. The background segmentation approach proposed here relies on an adapted PNN component to both classify the pixels and to store the model of the background within its weights. To achieve the functionality needed by a probabilistic video object segmentation algorithm, the adapted PNN component has been extended and combined with a Winner-Take-All(WTA) neural network. This resulted in a fairly complex solution with some unique properties. Namely, the proposed solution is a truly unsupervised classifier, requiring no training set and it is capable of on-line learning. To the best of our knowledge, this is the first PNN-based framework to achieve these properties, despite the use of PNN classifiers in myriad application domains [10] [20] [21] [22] [23]. The proposed neural-network



(a) $\sigma = 12$ (b) $\sigma = 24$

Fig. 2. Plots of Parzen estimators for different values of "smoothing parameter".

is referred to as Background Modeling Neural Network (BNN) since it is suitable to serve both as a statistical model of the background at each pixel position in the video sequences and highly parallelized background subtraction algorithm. As single BNN is used to model the pixel process and classify the pixel at a single pixel location l.

The basic idea that forms the basis of all probabilistic background modeling and video object 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:

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 neural network implementation proposed here this translates into the number of patterns stored, i.e. the number of neurons used per pixel.

The structure of BNN, shown in Fig. 3, has three distinct subnets, corresponding to each of the tasks enumerated above: *classification*, *activation* and *replacement*. The classification subnet is the adapted PNN discussed above. It is a central part of BNN shown in Fig. 3. The classification subnet contains four layers of neurons annotated at the far right of the Fig. 3. Input neurons of this network simply map the inputs of the network, which are the values of the features for a specific pixel. Each input neuron is connected to all pattern neurons. The output of the pattern neurons is a nonlinear function of Euclidean distance between the input of the network and the stored pattern for that specific neuron. The only parameter of this subnet is the smoothing parameter (σ) of the Parzen estimator, discussed previously. The output of a single pattern neuron corresponds to the value of a single Gaussian of the PDF estimation for the observed pixel value. Fig. 4 shows the structure of a pattern neuron of classification subnet. The

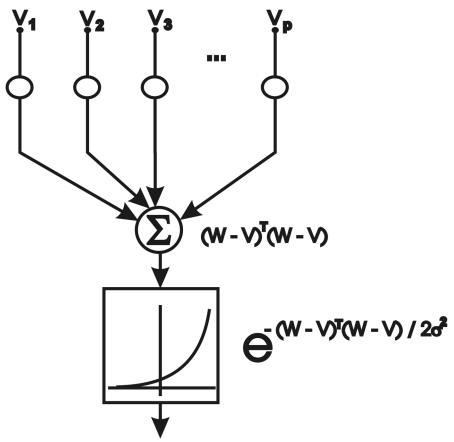


Fig. 4. Pattern neuron of PNN.

output of the summation units of the classification subnet is the sum of their inputs. The subnet has two summation neurons, each of them connected to all pattern neurons. The output values of the summation neurons correspond to initial Parzen estimates of joint probabilities $p_l(b, v)$ and $p_(f, v)$ for the pixel value observed (v). These estimates are input to the last (output) layer, containing a single neuron. The final output of the network is a binary value indicating whether the pixel corresponds to foreground (output high) or background (output low), i.e. the result of the comparison in (2).

A. Classification Subnet

1) Topology and Learning: The topology of the classification subnet is that of a PNN, as is the way in which the patterns

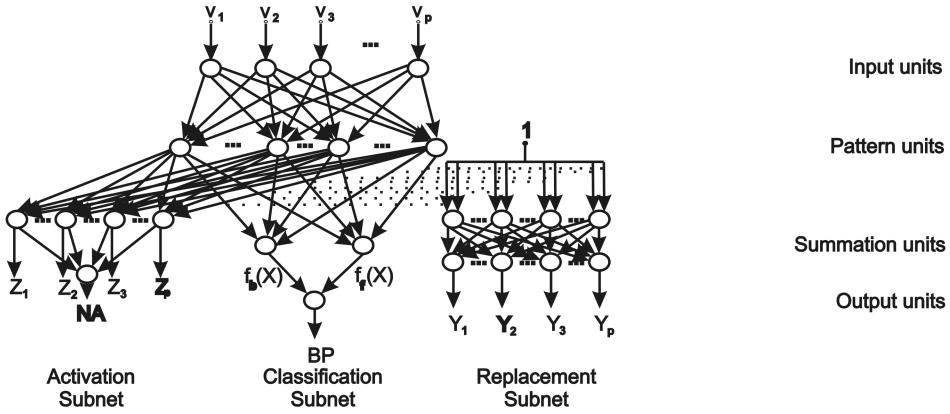


Fig. 3. Structure of Background Modeling Neural Network.

learnt are stored in the network. We discuss these briefly and refer the reader to [7] [8] for a more in depth discussion. In an PNN the patterns are stored in the weights connecting the input neurons and the pattern neurons. Originally, each pattern neuron corresponded to a single training pattern, and the weights of the connections between the input neurons and the pattern neuron were set to the values of the elements of the feature vector. This is inefficient and various clustering methods have been employed to reduce the number of patterns needed. The clustering method used in BNN is as proposed by Specht in [8]. It requires no external procedure is to determine whether a training pattern corresponds to a certain cluster. The pattern is simply fed to the network and if the output value of a pattern neuron exceeds a predefined threshold, the pattern is within the cluster covered by the pattern neuron. If no neuron achieves significant activation, the network is regarded as inactive and a pattern neuron is assigned to the pattern. It becomes a new cluster center. This procedure allows the network to adapt to the time-varying environment. In BNN all the values in the pixel processes are considered training patterns and the network undergoes permanent adaptation, i.e the training of the network is done on-line.

In the classification subnet of BNN, the weights between the pattern and summation neurons are used to store the the prior probabilities inferred for the pattern neuron value ($p_l(f|v)$ and $p_l(b|v)$). Since these values are unknown, rules were formed which allow the BNN to estimate them based on the observed parts of a pixel process and the frequency of specific feature values observed. The weights of these connections are updated with each new value of a pixel at a certain position received (i.e. with each frame), according to the following recursive equations:

$$W_{ib}^{t+1} = f_c((1 - \frac{\beta}{N_{pn}}) * W_{ib}^{t} + MA^{t}\beta) \tag{4}$$

$$W_{if}^{t+1} = 1 - W_{ib}^{t+1} \tag{5}$$

where W_{ib}^{t} is the value of the weight between the i-th pattern neuron and the background summation neuron at time t, W_{if}^{t} is the value of the weight between the i-th pattern neuron and the foreground summation neuron at time t, β is the learning rate, N_{pn} is the number of the pattern neurons of BNN, f_c is a clipping function defined by (6) and MA^t indicates the neuron with the maximum response (activation potential) at frame t, according to (7).

$$f_c(x) = \begin{cases} 1, & x > 1 \\ x, & x \leq 1 \end{cases} \tag{6}$$

$$MA^t = \begin{cases} 1, & \text{for neuron with maximum response;} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Equations (4) and (5) express the notion that whenever an instance pertinent to a pattern neuron is encountered, the probability that that pattern neuron is activated by a feature value belonging to the background is increased. Naturally, if that is the case, the probability that the pattern neuron is excited by a pattern belonging to foreground is decreased. Conversely, the more seldom a feature value corresponding to a pattern neuron is encountered, the more likely it is that the patterns represented by it belong to foreground objects. Since the PDF value of a single pattern is increased, while all the others are decreased, the decay rate is set to a value smaller than the increase rate by a factor equal to the number of stored patterns (pattern neurons). By adjusting the learning rate (β), it is possible to control the speed of the learning process.

2) Convergence of the Learning Process: Let T_o and T_{no} denote the number of frames in the sequence in which certain feature value of a pixel is observed and the number of frames in which it is not observed, respectively. Naturally, the overall number of frames in the sequence T is:

$$T = T_{no} + T_o \tag{8}$$

If a simplifying assumption is made, that the feature value is not observed for the first T_{no} frames of the sequence and then observed for T_o frames, the weights for the pattern neuron corresponding to the feature value are determined by equations (4)-(7):

$$W_{ib}^{T} = (1 - \frac{\beta}{N_{pn}})^{T_{no}} \cdot W_{ib}^{0} + T_o\beta \tag{9}$$

$$W_{if}^T = 1 - W_{ib}^T \tag{10}$$

where W_{ib}^0 corresponds to the initial weight set when the pattern is first observed. The initial weight has low value ($W_{ib}^0 \ll 1$), to indicate that it is not likely that the value observed for the first time corresponds to a background pixel. In addition, the values of the learning parameter are between 0 and 1. Observe that:

$$\beta \ll N_{pn} \tag{11}$$

Therefore:

$$W_{ib}^T \approx T_o \beta \tag{12}$$

Thus, if a pixel value is encountered in $\frac{1}{\beta}$ consecutive frames it will be classified as background with maximum support. On the other hand the confidence that a feature value belongs to the background will decay from the maximum to $(1-\frac{\beta}{N_{pn}})^{T_{no}}$ if it is not encountered for T_{no} frames.

B. Activation and Replacement Subnets

The adaptation of the classification subnet requires the BNN to be able to detect the state of low activation of all the neurons in the net. This indicates that the feature value fed to the network is not within the clusters stored and that the feature value should be stored in the network weights as a new cluster center. The activation part of BNN is concerned with the detection of this state. In addition, when the new value is to be stored, the network must be able to decide which pattern neuron's weights are to be replaced with new ones. This is the function of the replacement subnet.

The activation and replacement subnets are WTA neural networks. A WTA network is a parallel and fast way to determine minimum or the maximum of a set of values. In particular, these subnets are extensions of one-layer feedforward MAXNET (1LF-MAXNET) proposed in [9].

To detect the state of low activation in BNN the activation subnet determines which of the neurons of the network has maximum activation (output) and whether that value exceeds a threshold provided as a parameter to the algorithm. If it does not, the BNN is considered inactive and new cluster center learning process is initiated. If the network is inactive, the pixel is considered to belong to a foreground object, since this is a value that has not been present in the background model.

The first layer of the activation network has the structure of a 1LF-MAXNET network and a single neuron is used to indicate whether the network is active. The output of the neurons of the first layer of the network can be expressed in the form of the following equation (see Fig. 5):

$$Y_j = X_j \times \prod_{i=1}^{P} \{F(X_j - X_i | i \neq j)\} \tag{13}$$

where:

$$F(z) = \begin{cases} 1, & \text{if } z \geq 0; \\ 0, & \text{if } z < 0; \end{cases} \tag{14}$$

As the (13) and (14) indicate, the output of the first layer of the activation subnet will differ from 0 only for the neurons with maximum activation and will be equal to the maximum activation. In Fig. 3 these outputs are indicated with $Z_1, , Z_P$. The structure of a processing neuron of 1LF-MAXNET is shown in Fig. 5. A single neuron in the second layer of the

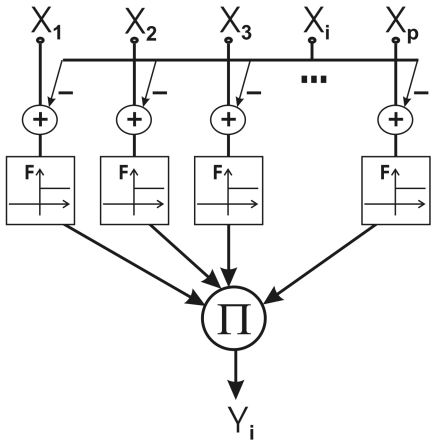


Fig. 5. Processing neuron of a 1LF-MAXNET structure.

activation subnet is concerned with detecting whether the BNN is active or not and its function can be expressed in the form of the following equations:

$$NA = F(\sum_{i=1}^{P} Z_i - \theta) \tag{15}$$

where F is given by (14) and θ is the activation threshold, which is provided to the network as a parameter. Finally, the replacement subnet in Fig. 3 can be viewed as a separate neural net with the unit input. However, it is inextricably related to the classification subnet since each of the replacement subnet first-layer neurons is connected with the input via synapses that have the same weight as the two output synapses between the pattern and summation neurons of the classification subnet. Each pattern neuron has a corresponding neuron in the replacement net. The function of the replacement net is to determine the pattern neuron that minimizes the criterion for cluster center replacement, expressed by the following equation:

$$replacement\_criterion = W_{if}^t + |W_{ib}^t - W_{if}^t| \tag{16}$$

The criterion is a mathematical expression of the idea that those patterns that are least likely to belong to the background and those that provide least confidence to make the decision should be the first to be eliminated from the model.

The neurons of the first layer calculate the negated value of the replacement criterion for the pattern neuron they correspond to. This inversion of the sign is done to enable the use of 1LF-MAXNET component to detect the lowest value of the criterion. The second layer of the network is a 1LF-MAXNET that yields non-zero output corresponding to the pattern neuron to be replaced.

C. Hardware implementation considerations

Suitability of the proposed solution for hardware implementation is a primary concern in our research.

Since each neuron within a layer of the BNN is able of performing its function in parallel, the proposed approach is parallelized on a sub-pixel level. More precisely, it is parallel on the level of a single pattern stored for a pixel, since the pattern neurons of the classification subnet are able to perform their calculations in parallel.

The presence of the WTA networks eliminates the need for sorting operations employed both in the hybrid approach of Li *et al.* and MoG. In addition, there is no need to perform any operations on the whole frame, such as histogram extraction for adaptive thresholding performed in the approach of Li *et al.* [24], which limit the extent to which the approach can be parallelized and make the speed of segmentation dependent on the size of the frame. No publications have been identified dealing with parallel or hardware implementations of MoG and the approach of Li *et al.*. However, Li *et al.* report [4] that their approach can achieve processing speed of 15 fps for 160×120 pixel large frames and 3 fps for 320×240 pixel large frames when run on a 1.7 GHz Pentium CPU. For MoG containing 5 Gaussians per pixel, processing rate of 11 to 13 fps (frame size 160×120 pixels), on an SGI O2 with an R10000 processor, has been reported [3].

The speed of the segmentation of the proposed approach, in a parallel hardware-implementation, does not depend on the size of the frame. The delay of the network (segmentation time) corresponds to the time needed by the signal to propagate through the network and time required to update it. In a typical FPGA implementation this can be done in less than 20 clock cycles, which corresponds to a 2 ms delay through the network, for a FPGA core running at 10 MHz clock rate. Thus, the networks themselves are capable of achieving a throughput of some 500 fps, which is more than sufficient for real-time segmentation of video sequences.

BNN is clearly suitable to serve as basis for efficient hardware implementation.

V. Experiments and Results

To evaluate the performance of the neural network a sequential PC based implementation has been developed. Previously, experiments have been conducted on a set of sequences pertinent to marine surveillance [6]. Here, a set of diverse sequences containing complex background conditions, provided by Li *et al.* [4] and publicly available at *http://perception.i2r.a-star.edu.sg*, was used. The results of the segmentation were evaluated both qualitatively and quantitatively, using a set of ground truth frames provided by the same authors for the different sequences. To evaluate the performance of the approach, a well-known probabilistic modelling approach MoG [3] and the approach of Li *et al.* [4], have been implemented and the segmentation results of different algorithms compared. Same pixel feature, namely its intensity value was used for all approaches. Since all three are general in terms of features used, the intent is to compare base-line modelling and classification ability rather than explore the problem of selecting the best features in order to achieve best segmentation results. Note however, that use of different features can affect the segmentation quality and that employing different features should be considered as a possibility both in applications and as a venue to further explore the quality of the proposed method.

TABLE I
Learning rate used in experiments.

	CAM	SW	FT	WS	MR
β	0.05	0.003	0.01	0.01	0.003
	LB	SC	AP	BR	SS
β	0.01	0.01	0.03	0.003	0.05

The neural networks used in the experiments are fairly simple. The simulation application implements BNNs containing 30 pattern neurons in their classification subnets. With the additional two summation and one output neuron required in the classification subnet (see Fig. 3), the total number of neurons in this part of BNN is 33. The input neurons of the classification shown in Fig. 3 just map the input to the output and need not be implemented as such. The number of neurons required in the activation and replacement subnets is determined by the number of pattern neurons of the classification subnet. These two subnets attribute for additional 31 neurons in the activation subnet and 60 processing units in the replacement subnet. Thus, the total number of neurons in a single BNN used is 124. A single BNN is used to model the background at a single pixel.

The learning rate (β) varied from sequence to sequence between three different settings (0.05, 0.01 and 0.003) as shown in Table I. The learning rates have been set based on the observed speed of motion of the objects in the foreground and rate of changes in the background. Larger learning rates enable the network to learn the changes corresponding to background faster but lead to faster absorption of stationary foreground objects by the background model. Lower learning rates make the network slower to adapt to sudden changes in the background (e.g. due to switching of lights) but will make the model less prone to errors due to absorption of stationary foreground objects.

The smoothing parameter (σ) for the classification subnet used was set to 7 (a value approximately twice the size standard deviation of the intensity values for a single pixel) . The activation threshold (θ) of the activation subnet was set to 0.5, meaning that the values further than $\approx 1.18\sigma$ form the pattern neuron weights were deemed outside the scope of the cluster covered by that particular neuron.

The results for MoG were obtained for a mixture containing 30 Gaussians, to ensure a fair comparison. While the authors suggest the use of 3-5 Gaussians in the mixture to achieve real-time performance [3], they speculated that a larger number of Gaussians would lead to better segmentation results. The initial value of the deviation for MoG was set to the value of the smoothing parameter of BNN (7) while the threshold selecting the number cases to be covered by Gaussians used in the decision process was set to 0.5. A Gaussian covered the values within 2.5 standard deviations of the mean, as suggested in [3]. The learning rates for the MoG were set to the same values as for the BNN.

In the experiments with the approach of Li *et al.* the number of pixel intensity values and pixel intensity co-occurrence values was adopted from [17]. No binning of the intensity values was performed, while bins of width 4 were used for co-occurrence features. In addition, to ensure a fair comparison, the probabilistic background model has been updated based on the initial segmentation results, before the application of morphological processing, as it was done for the other two approaches. The learning rates used for the experiments were those suggested in [4]. While Li *et al.* suggest that the same learning rates should be used for their approach and MoG [4], they do not consider the interplay of two different learning rates used on the two levels of their algorithm. Thus, this suggestion was not followed in the experiments performed here.

Morphological operations such as morphological closing and opening as well as connected components algorithm and the elimination of small objects have been used to enhance the segmentation results, as it was done in [4]. All the segmentation results, along with the binaries used for segmentation and MATLAB scripts to perform the morphological processing and extract the statistics used for quantitative comparison can be found at `http://mlab.fau.edu`.

A possible alternative to morphological processing could be the use of a still image segmentation algorithm, such as that proposed by Blekas *et al.* [25] to enhance the segmentation results. The resulting algorithm would have the benefit of exploiting spatial information. To improve the performance, still image segmentation could be localized to regions already detected as foreground.

A. Qualitative Results

Qualitative evaluation of the segmentation results is performed by visual inspection. In this section a number of representative frames from the test sequences is presented. We discuss the nature and the causes of complexity of background changes. The discussion is limited and the reader is referred to [4] for a more in-depth treatment.

Ten testing sequences were obtained in several different environments. Rather than following the classification originally used in [4], they are grouped here based on the sources of complexity in background variation, pertinent to each environment. Three groups of sequences (environments) are identified:

1) Outdoor environments.
2) Small indoor environments.
3) Large (public) indoor environments.

The sources of complexity in the sequences obtained in outdoor environments are usually due to objects moved by wind (e.g. fig3 or waves) and illumination changes due to changes in cloud cover. For small indoor environments, such as offices, the source of complexity related mostly to objects such as curtains or fans moving in the background or screens flickering. The illumination changes are mostly due to switching lights on and off. Large public indoor environments (e.g. subway stations, airport halls, shopping centers etc.) are characterized by lighting distributed from the ceiling and presence of secular surfaces, inducing complex shadow and glare effects. In addition, these spaces can contain large moving objects such as escalators and elevators.

Initial results presented in [6] are concerned with outdoor sequences with background containing water surfaces as well as objects undergoing motion due to wind. To evaluate the segmentation results for the outdoor environments further, four sequences were used. The first sequence is of a campus environment (CAM), showing vehicles and pedestrians moving along a road in front of a thicket of fig3 moving rapidly in the wind. Second is that of a sidewalk (SW) with pedestrians moving along. The complexity of the background in the third sequence (FT) is due to a water fountain. The fourth (WS) is a sequence of a person at walking at a waterfront, and the complexity is due to the water-surface in the background. The proposed algorithm was able to cope with complex background variation in all these sequences. Representative frames and corresponding segmentation results are given in Figs. 6,7,8 and 9, for the sequences CAM, SW, FT and WS, respectively. The figures show the original frame, the segmentation result obtained using BNN, ground truth frame, segmentation result of MoG and segmentation result for the model of the background proposed by Li *et al.*, from left to right. The ground truths are manually segmented frames. All the images referred to in this subsection have the same format.

Two sequences are used to test the performance for small indoor environments. The first was captured in a meeting room (MR) with the curtain moving in the background. The second (LB) was taken in the lobby of the office building, with the lights switching on and off. Representative frames for the two sequences are shown in Figs. 10 and 11.

Four sequences pertinent to large indoor environments were used. They were taken in a shopping center (SC), an airport (AP), a buffet restaurant (BR) [2] and a subway station (SS). They illustrate the capability of the approach to cope with shadow effects. The subway station sequence contains moving escalators in the background. Segmentation results for these sequences are illustrated in Figs. 12,13,14 and 15. The algorithm was able to learn the behavior of the escalators in the SS sequence and absorb them into the background. The algorithm does not incorporate shadow cancellation and has in some cases segmented the shadows as foreground objects as shown in 14(b). A possible solution to this problem lies in the use of features able to cope with these effects, as are features in gradient domain proposed in [26]. In addition, a more sophisticated approach, such as that proposed by Gu *et al.* [27], could be applied to the segmented regions, to remove shadows at the higher levels of processing.

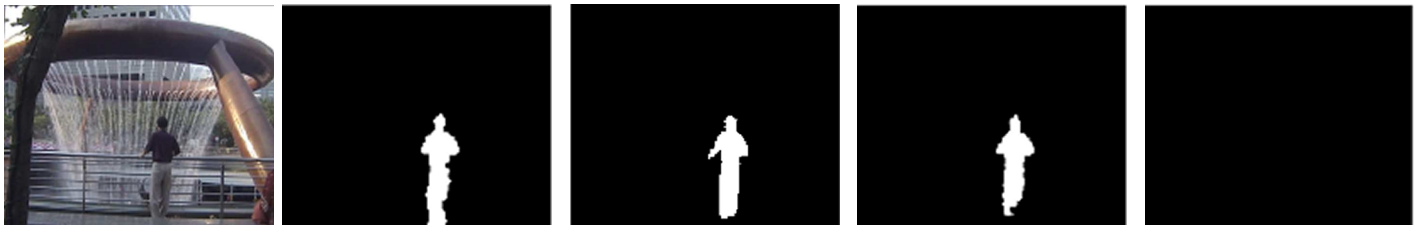
An additional weakness of the the proposed algorithm is the tendency to incorporate foreground objects, that stop moving for extended periods of time, into the background. This is a weakness of all background modeling segmentation approaches, stemming from the basic definitions of background and foreground stated above. A possible solution to this problem is the use of top-down information from higher-level object tracking and recognition algorithms.

(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 6. Segmentation of the campus sequence (CAM).

(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 7. Segmentation of the buffet sidewalk sequence (SW).



(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 8. Segmentation of the water fountain sequence (FT).

(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 9. Segmentation of the water surface sequence (WS).

(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 10. Segmentation of the meeting-room sequence (MR).

(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 11. Segmentation of the office-building lobby sequence (LB).

(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 12. Segmentation of the shopping center sequence (SC).

(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 13. Segmentation of the airport sequence (AP).

(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 14. Segmentation of the buffet restaurant sequence (BR).

(a) Original frame. (b) BNN segmentation result. (c) Ground truth. (d) MoG result. (e) Li *et al.* result.

Fig. 15. Segmentation of the subway-station sequence (SS).

TABLE II
SIMILARITY MEASURE VALUES FOR EACH TEST SEQUENCE.

	CAM	SW	FT	WS	MR
BNN	0.5256	0.6216	0.4636	0.7540	0.7368
MoG	0.0757	0.5861	0.6854	0.7948	0.7580
Li et al.	0.1596	0.1032	0.0999	0.0667	0.1841
	LB	SC	AP	BR	SS
BNN	0.6276	0.5696	0.3923	0.4779	0.4928
MoG	0.6519	0.5363	0.3335	0.3838	0.1388
Li et al.	0.1554	0.5209	0.1135	0.3079	0.1294

B. Quantitative Evaluation

For each of the ten test sequences we calculate a measure of the segmentation accuracy following the methodology used in [17]. If D is a detected(segmented) region and G the corresponding ground truth, then the similarity measure between these two regions is defined as:

$$S = \frac{D \cap G}{D \cup G} \tag{17}$$

The similarity of the regions (S) reaches the maximum value of 1 if they are the same. Otherwise, it varies between 1 and 0 according to how similar the regions are. S is the measure of the overall misclassification. The values of S obtained for test sequences along with the values provided by Li *et al.* for the MoG and their own approach are given in Table II. The average values obtained for the three approaches are 0.566, 0.494 and 0.184 for BNN, MoG and the approach of Li *et al.*, respectively.

As Table II indicates, proposed approach achieved better segmentation results than the MoG containing the same number of Gaussians as there are pattern neurons in the BNN. The approach proposed by Li *et al.* performed poorly when only intensity values and their co-occurrences are used for segmentation. This is quite different than the result reported in [4]. The discrepancy is probably due to the fact that the authors used higher-level features and first order moments in addition to intensity values to classify pixels.

VI. CONCLUSION

A novel background modeling and subtraction approach for video object segmentation in complex sequences has been proposed. The proposed method is probabilistic and relies on a neural network to achieve estimation of required PDFs and segmentation. New *Background modeling Neural Network* (BNN) architecture has been proposed and rules for adaptation of its weights have been formulated. The network is a truly unsupervised classifier, differing from previously published approaches. The algorithm is parallel on a sub-pixel level. Of the published segmentation approaches, it is most suitable for an efficient hardware implementation.

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 pure probabilistic approach MoG. For several sequences MoG featuring 30 Gaussians achieved better results (FT, WS, MR and LB). This result indicates that the proposed approach could benefit from introduction of adaptive kernel width and center. Both MoG and the proposed approach performed significantly better than the hybrid model-based approach of Li *et al.*, when the pixel intensity values are used as the basis for segmentation.

The approach is independent of the features used to achieve segmentation and use of features other than intensity values should be explored to enhance the segmentation results, especially in terms of shadow suppression. 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. Such top-down control could be used to cope with the problem of foreground objects being absorbed by the background.

ACKNOWLEDGMENT

The authors would like to thank Spyros Magliveras for his support and valuable suggestions.

REFERENCES

[1] I. Haritaoglu, D. Harwood, and L. Davis, "W4 real-time surveillance of people and their activities," in *IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, pp. 809-830*, 2000.

[2] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *In Proceedings of IEEE Int'l Conf. On Computer Vision, pp. 255-261*, 1999.

[3] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," in *IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, pp. 747-757*, 2000.

[4] L. Li, W. Huang, I. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," in *IEEE Trans. Image Processing, vol. 13, pp. 1459-1472*, 2004.

[5] D. Socek, D. Culibrk, O. Marques, H. Kalva, and B. Furht, "A hybrid color-based foreground object detection method for automated marine surveillance," in *Proc. of the Advanced Concepts for Intelligent Vision Systems Conference (ACIVS 2005)*, 2005.

[6] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "A neural network approach to bayesian background modeling for video object segmentation," in *Proc. of the International Conference on Computer Vision Theory and Applications (VISAPP'06)*, 2006.

[7] D. F. Specht, "Probabilistic neural networks," in *Neural Networks, vol. 3, pp. 109-118*, 1990.

[8] ——, "A general regression neural network," in *IEEE Trans. Neural Networks, pp. 568-576*, 1991.

[9] H. K. Kwan, "One-layer feedforward neural network fast maximum/minimum determination," in *Electronics Letters, pp. 1583-1585*, 1992.

[10] M. R. Azimi-Sadjadi, W. Gao, T. H. V. Haar, and D. Reinke, "Temporal updating scheme for probabilistic neural network with application to satellite cloud classification-further results," in *IEEE Trans. Neural Networks, vol. 12, pp. 1196-1203*, 2001.

[11] A. Doulamis, N. Doulamis, K. Ntalianis, and S. Kollias, "An efficient fully unsupervised video object segmentation scheme using an adaptive neural-network classifier architecture," in *IEEE Trans. On Neural Networks, vol. 14, pp. 616-630*, 2003.

[12] G. Pajares, "A hopfield neural network for image change detection," in *IEEE Trans. Neural Networks, vol. 17, pp. 1250-1264*, 2006.

[13] K. P. Karmann and A. von Brandt, "Moving object recognition using an adaptive background memory," in *Timevarying Image Processing and Moving Object Recognition, 2, pp. 297-307*. Elsevier Publishers B.V., 1990.

[14] T. Boult, R. Micheals, X.Gao, P. Lewis, C. Power, W. Yin, and A. Erkan, "Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets," in *Proc. of IEEE Workshop on Visual Surveillance, pp. 48-55*, 1999.
[15] T. Ellis and M. Xu, "Object detection and tracking in an open and dynamic world," in *Proc. of the Second IEEE International Workshop on Performance Evaluation on Tracking and Surveillance (PETS'01)*, 2001.
[16] L. Ya, A. Haizhou, and X. Guangyou, "Moving object detection and tracking based on background subtraction," in *Proc. of SPIE Object Detection, Classification, and Tracking Technologies, pp. 62-66*, 2001.
[17] L. Li, W. Huang, I. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proc. of the Eleventh ACM International Conference on Multimedia (MULTIMEDIA'03), pp. 2-10*, 2003.
[18] E. Parzen, "On estimation of a probability density function and mode," in *Ann. Math. Stat., Vol. 33, pp. 1065-1076*, 1962.
[19] A. M. Mood and F. A. Graybill, *Introduction to the Theory of Statistics.* Macmillan, 1962.
[20] C. Kramer, B.Mckay, and J. Belina, "Probabilistic neural network array architecture for ecg classification," in *Proc. Annu. Int. Conf. IEEE Engineering Medicine Biology, 17, pp. 807808*, 1995.
[21] M. T. Musavi, K. H. Chan, D. M. Hummels, , and K. Kalantri, "On the generalization ability of neural-network classifier," in *IEEE Trans. Pattern Anal. Machine Intell., vol. 16, pp. 659663*, 1994.
[22] P. P. Raghu and B. Yegnanarayana, "Supervised texture classification using a probabilistic neural network and constraint satisfaction model," in *IEEE Trans. Neural Networks, vol. 9, pp. 516522*, 1998.
[23] R. D. Romero, D. S. Touretzky, and G. H. Thibadeau, "Optical chinese character recognition using probabilistic neural netwoks," in *Pattern Recognit., vol. 3, pp. 12791292*, 1997.
[24] L. Rosin, "Thresholding for change detection," in *Proc. of the Sixth International Conference on Computer Vision (ICCV'98)*, 1998.
[25] K. Blekas, A. Likas, N. P. Galatsanos, and I. E. Lagaris, "A spatially constrained mixture model for image segmentation," in *IEEE Trans. Neural Networks, vol. 16, pp. 494-498*, 2005.
[26] S. Chien, Y. Huang, B. Hsieh, S. Ma, and L. Chen, "Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques," in *IEEE Trans. on Multimedia, vol. 6, pp. 732-748*, 2004.
[27] X. Gu, D. Yu, and L. Zhang, "Image shadow removal using pulse coupled neural network," in *IEEE Trans. Neural Networks, vol. 16, pp. 692-698*, 2005.

Dubravko Ćulibrk received a B.Eng. degree in automation and system control as well as a M.Sc. degree in computer engineering from the University of Novi Sad, Novi Sad, Serbia, in 2000 and 2005, respectively. In 2006 he received a Ph.D. degree in computer engineering from Florida Atlantic University, Boca Raton, Florida, USA. His research interests include video and image processing, computer vision, neural networks and their applications, cryptography, hardware design and evolutionary computing.

Oge Marques is an Assistant Professor in the Department of Computer Science and Engineering at Florida Atlantic University in Boca Raton, Florida. He received his B.S. degree in Electrical Engineering from *Centro Federal de Educação Tecnológica do Paraná* (CEFET-PR) in Curitiba, Brazil, a Master's degree in Electronic Engineering from Philips International Institute of Technological Studies in Eindhoven, The Netherlands, and a Ph.D. degree in Computer Engineering from Florida Atlantic University. Dr. Marques's research and publications have been focused on: image processing, analysis, annotation, search, and retrieval; video processing and analysis; and secure multimedia communications. He is a member of ACM, IEEE, IEEE Computer Society, and the honor societies of Phi Kappa Phi and Upsilon Pi Epsilon.

Daniel Socek is a research assistant Department of Computer Science and Engineering at Florida Atlantic University (FAU). He is affiliated with the Center for Coastline Security Technology and the Center for Cryptology and Information Security at FAU. He graduated from University of Nebraska-Lincoln in 2000 with the B.Sc. degree in computer science, and received an M.Sc. in mathematics and Ph.D. degree in computer science from Florida Atlantic University in 2002 and 2006, respectively. His current research interests include multimedia security, biometrics-based authentication, digital image and video processing, and applications of neural networks. For his outstanding academic and research performance, he was awarded the Graduate Fellowship for Academic Excellence and the Dr. Daniel B. and Aural B. Newell Doctoral Fellowship in 2003 and 2004, respectively. He worked and served as a consultant for various IT companies such as IBM, Panasonic, Matsushita Electric Industrial, and Avaton/Crypton. He is also a student IEEE member.

Hari Kalva joined the Department of Computer Science and Engineering at Florida Atlantic University as an Assistant Professor in August 2003. Prior to that he was a consultant with Mitsubishi Electric Research Labs, Cambridge, MA, where he worked different projects including MPEG-2 to MPEG-4 realtime video transcoding. He was a co-founder and the Vice President of Engineering of Flavor Software, a New York company founded in 1999, that developed MPEG-4 based solutions for the media and entertainment industry. He is an expert on digital audio-visual communications systems with over ten years of experience in multimedia research, development, and standardization. He has made key contributions to the MPEG-4 Systems standard and also contributed to the DAVIC standards development. His research interests include pervasive media delivery, content adaptation, video compression, and communication. He has over a two-dozen published papers and four patents (six pending) to his credit. He is the author of one book and co-author of five book-chapters.

Borko Furht is Chairman and Professor in the Department of Computer Science and Engineering at Florida Atlantic University (FAU) in Boca Raton, Florida.His research interests include multimedia systems and applications, video processing, wireless multimedia, multimedia security, video databases, and Internet engineering. He has published more than 20 books and about 200 scientific and technical papers, and holds 2 patents.