Video Object Segmentation with Multivalued Neural Networks

R.M. Luque
Dept. of Computer Science
ETSI Informática
University of Málaga, 29071 Málaga, Spain
rmluque@lcc.uma.es

D. López-Rodríguez, E. Mérida-Casermeiro
Dept. of Applied Mathematics
ETSI Informática
University of Málaga, 29071 Málaga, Spain
{dlopez,merida}@ctima.uma.es

E.J. Palomo
Dept. of Computer Science
ETSI Informática
University of Málaga, 29071 Málaga, Spain
ejpalomo@lcc.uma.es

Abstract

The aim of this work is to present a segmentation method to detect moving objects in video scenes, based on the use of a multivalued discrete neural network to improve the results obtained by an underlying segmentation algorithm. Specifically, the multivalued neural model (MREM) is used to detect and correct some of the deficiencies and errors off the well-known Mixture of Gaussians algorithm. Experimental results, using video scenes publicly available from the Internet, show an increase of the visual quality of the segmentation, that could improve for subsequent analysis phases, such as object tracking or behavior studies.

1 Introduction

The aim of moving object segmentation is to separate pixels corresponding to foreground from those corresponding to background. The increasing resolution of video sequences, and continuous advances in video capture and transmission technology, make segmentation a complex task.

Some works [3, 9, 10] propose the creation of a background approximation by averaging the images over time. This process of modeling the background by comparing several frames in a video sequence, usually referred to as background subtraction, is one of the standard methods for video object detection. These methods are quite effective in scenes where objects move continuously, whereas they lack robustness when many moving objects are present in the sequence, particularly if they move slowly.

Wren et al. [18] used a multiclass statistical representation based on Gaussian distributions, in which the background model is a single Gaussian distribution per pixel. A modified version modeling each pixel as a mixture of Gaussians (MoG) was proposed by Stauffer and Grimson [17]. This statistical approach is robust in scenes with many moving objects and light changes, and it is one of the most cited techniques in the literature.

The aim of this work is to present the use of a multivalued neural model (MREM) [12] to enhance the segmentation result obtained by other specialized algorithms, by avoiding some of the undesirable effects of them. Particularly, we propose the hybridization of this MREM model and the Mixture of Gaussians algorithm, since the neural model is able to optimize the segmentation obtained by the latter method.

One of the advantages of using neural networks for decision and optimization problems [7] is that all process units (neurons) compute the solution to the problem in parallel. This means that more complex problems can be solved, due to the use of more efficient algorithms.

Another interesting feature of the multivalued neural model studied in this work is its ability to represent non-numerical classes or states, very useful when dealing with image segmentation problems, in which pixel states are usually defined with qualitative labels: {foreground, background, shadow}.

In addition, since many of the segmentation algorithms are pixel-oriented, that is, they study the most probable class of each pixel separately, the use of a neural network helps to introduce some information on the neighborhood of each pixel to get a smooth segmentation.

The remainder of this paper is structured as follows: in

978-0-7695-3326-1/08 $25.00 © 2008 IEEE
DOI 10.1109/HIS.2008.130

section 2, the problems of using a pixel-level segmentation technique are presented, along with the description of the Mixture of Gaussians algorithm. Later, in section 3, the multivalued neural model MREM is described, as well as its use to improve the segmentation results obtained by Mixture of Gaussians. Some experimental results are shown in section 4. To end with, in section 5, some conclusions and future work regarding the topic of this work are presented.

2 The Problem of Pixel Segmentation

Many works related to background subtraction are based on modeling each pixel of the image by using a statistical model [18, 17]. The first objective of these methods is to extract moving objects from the background.

One of the most cited techniques is the Mixture of Gaussians (MoG) introduced by Stauffer and Grimson [17], in which each pixel in the scene is modeled by means of a weighted sum (mixture) of K Gaussian distributions. This algorithm is considered as a pixel-level method. Different Gaussians are assumed to represent different regions of the RGB color space. Let us briefly explain this method.

The probability that a certain pixel has a value $X_t = (R_t, G_t, B_t)$ at frame t, can be written as

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

where $\omega_{i,t}$ is an estimate of the weight of the i-th Gaussian in the mixture at time t, $\mu_{i,t}$ and $\Sigma_{i,t}$ are the mean value and covariance matrix of the i-th Gaussian in the mixture at time t, and η is a Gaussian probability density function:

$$\eta(X, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)} \quad (1)$$

where n is the dimension of the input space, that is, the number of components of X (in this case, $n = 3$), and $|\Sigma|$ represents the determinant of matrix Σ.

By assuming that, for each distribution, red, green, and blue values are independent and have the same variances, $\Sigma_{i,t} = \sigma_i^2 I$, an approximation of the posterior probability $\eta(X_t, \mu_i, \Sigma_i)$ is used to reduce the time complexity of the algorithm.

We say that pixel color, X_t, at time t, is represented by the i-th Gaussian distribution in the mixture (or, for simplicity, that the distribution matches the pixel value) if the match value $M_{i,t} = \frac{|X_t - \mu_{i,t}|}{\sigma_{i,t}}$ verifies $M_{i,t} < 2.5$, that is, the current RGB vector of the pixel lies within 2.5 standard deviations of distance from the mean of the Gaussian distribution.

After a match is done, the weight of each distribution matching the pixel value is updated by using the equation

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha(M_{k,t})$$

for a given value of α, the learning rate, usually decreasing to small fixed value.

If none of the K distributions matches the current pixel value, the least probable distribution is replaced with a distribution with the current (R,G,B) value as its mean value, and initially high variance, and low prior weight. In our case, $\sigma = 25$ and ω is the less value of the weights of the distributions.

The μ and σ parameters for unmatched distributions remain the same. The parameters of the distribution which matches the new observation are updated as follows:

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho X_t$$

$$\sigma_{i,t}^2 = (1 - \rho)\sigma_{i,t-1}^2 + \rho(X_t - \mu_{i,t})^T(X_t - \mu_{i,t})$$

where

$$\rho = \frac{\alpha M_{i,t}}{\omega_{i,t}}$$

This implementation is faster than the previously proposed in [17].

After the update process, it is necessary to determine which of the Gaussians of the mixture is most likely produced by background processes. A distribution is deemed to be background with high probability if it occurs frequently (high ω) and does not vary much (low standard deviation σ). Therefore, the Gaussian distributions are sorted according to this criterion:

$$\frac{\omega_1}{\sigma_1} \geq \frac{\omega_2}{\sigma_2} \geq \ldots \geq \frac{\omega_K}{\sigma_K}$$

The first B distributions are considered to belong to the background model:

$$B = \arg\min_b (\sum_{k=1}^{b} \omega_k > T)$$

where T represents the minimum weight that a set of distributions must have to be considered as background. The rest of distributions are considered part of the foreground objects.

In this work, we have used $K = 3$ Gaussian distributions to model each pixel color space. These 3 distributions represent background, foreground objects and shadows.

In order to identify and remove moving shadows, we need to consider a colour model that can separate chromatic and brightness components. This can be done by comparing non-background pixels against the current background components. If the difference in both chromatic and brightness components is within some threshold, the pixel is considered as a moving shadow. We use an effective computational color model similar to the one proposed in [8] to fulfill these needs.

Despite the good segmentation results obtained by applying MoG in each scene, a large amount of spurious objects are detected, mostly, due to the fact that no relation with the neighborhood of each pixel is taken into account to obtain these objects in motion. Post processing methods such as morphological operators are used to avoid the disadvantages of this kind of pixel level techniques. However, the input parameters for these methods depend on the scene to analyze, therefore, the results will not be too satisfactory if these fixed parameters are not optimally adjusted. To solve this situation and improve the segmentation results, a new optimization technique, based on multivalued neural networks, is developed in this work.

3 Multivalued Neural Network to Improve Segmentation

In this section, the fundamentals of the Multivalued RECurrent Model (MREM) [12] are described. This discrete neural network is a generalization of Hopfield's model [6, 7] and other binary and multivalued models, such as SOAR [15] and MAREN [4].

3.1 The MREM Model

Let us consider a recurrent neural network formed by N neurons, where the state of each neuron $i \in \mathcal{I} = \{1, \ldots, N\}$ is defined by its output v_i taking values in any finite set $\mathcal{M} = \{m_1, m_2, \ldots, m_L\}$. This set does not need to be numerical. For example, $\mathcal{M} = \{\text{red}, \text{green}, \text{blue}\}$ or $\mathcal{M} = \{\text{background}, \text{foreground}, \text{shadow}\}$.

The vector V whose components are the corresponding neuron outputs, $V = (v_1, v_2, \ldots, v_n)$, is called state vector. Associated to each state vector, an energy function, similar to Hopfield's, can be defined:

$$E(V) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j} f(v_i, v_j, i, j) + \sum_{i=1}^{n} \theta_i(v_i) \quad (2)$$

where

- $W = (w_{i,j})$ is the synaptic weight matrix, expressing the connection strength between neurons.
- $f\colon \mathcal{M} \times \mathcal{M} \times \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ is the so-called similarity function, since $f(v_i, v_j, i, j)$ measures the similarity between the outputs of neurons i and j.
- $\theta_i\colon \mathcal{M} \to \mathbb{R}$ is the generalization of the biases $\theta_i \in \mathbb{R}$, present in Hopfield's model.

The aim of the network is to minimize the energy function (2), i.e., to achieve a stable state corresponding to a local (global, when possible) minimum of the energy function, which is usually identified with the objective function of the problem to solve.

The introduction of the similarity function f makes the network very versatile and usually causes a better representation of the problem at hand, see, for example, [14, 13, 11, 5]. It leads to a better representation of problems than other multivalued models, as SOAR and MAREN [4, 15], since in those models most of the information enclosed in the multivalued representation is lost by the use of the signum function that only produces values in $\{-1, 0, 1\}$.

Usually, the similarity function is defined only in terms of v_i and v_j, but, in this work, we have decided to introduce some information about pixel adjacency, by giving the indices of the pixels (i and j). This information was proposed in [14].

Many computational dynamics can be defined for this model, that is, several neuron updating schemes are available provided the versatility of the network. This is achieved by defining the input (synaptic potential) for neuron p, u_p, as the opposite of the energy increase when neuron p is updated, that is, $u_p = -(\Delta E)_p$. If $u_p > 0$, then the associated update reduces the value of the energy function. Otherwise, since no improvement is made by the update, it is not done, and the net has converged to a stable state.

3.2 MREM Applied to Segmentation

In this paper, we propose the use of MREM to enhance the segmentation obtained by other segmentation methods. Particularly, MREM is used to improve the results achieved by the Mixture of Gaussians method.

The pixel segmentation problem can be considered as an optimization problem that can be solved by MREM. In order to make this identification, we can define the set of possible neuron states as $\mathcal{M} = \{\text{background}, \text{foreground}, \text{shadow}\}$, such that v_i, the state of neuron i, represents the class the i-th pixel is assigned to.

Several measures of the quality of the segmentation can be defined [2]. In this work, we propose a new measure of the quality, which corresponds to the objective function of the associated optimization problem.

This objective function is defined in terms of the value $m_j(k)$ which measures the fuzzy membership of pixel j to class $k \in \{\text{background}, \text{foreground}, \text{shadow}\}$.

Typically, $m_j(k)$ is defined as:

$$m_j(k) = \eta(x_j, \mu_j(k), \Sigma_j(k))$$

where $\mu_j(k)$ and $\Sigma_j(k)$ are, respectively, the mean and the covariance matrix of the Gaussian distribution associated to class k at pixel j, and x_j is the (R,G,B) vector of pixel j at the current frame, and η is the Gaussian probability density function defined in Eq. (1).

Thus, the minimization problem can be stated as follows:

$$\min_{V \in \mathcal{M}^N} \sum_i \sum_{j \in \mathcal{N}_i} \left(1 - 2\delta_{v_i, v_j}\right) m_j(v_j) \tag{3}$$

where $\mathcal{N}_i$ represents the neighborhood of pixel i and $\delta_{x,y}$ is Kronecker's delta function. The use of a neighborhood for each pixel allows to obtain smooth segmentations.

The optimal solution to this problem is achieved when a pixel is assigned to the most probable class (in terms of the weighted membership value) of pixels in its vicinity. For example, if the sum of weighted membership values of pixels in class *foreground* is greater than that of the pixels in any of the two remaining classes, then the current pixel is assigned to the class *foreground*.

With the objective function given in (3), the obtained segmentation is smoother and presents less spurious objects than the original MoG algorithm.

The identification of this objective function with the energy function of MREM, given in (2), allows us to define the model parameters:

$$\begin{aligned} w_{i,j} &= \begin{cases} -2, & \text{if } j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases} \\ f(x, y, i, j) &= \left(1 - 2\delta_{x,y}\right) m_j(y) \\ \theta_i &\equiv 0 \quad \forall i \end{aligned}$$

For this problem, a parallel dynamics have been considered in which only one neuron is updated at each iteration. Every neuron computes, in parallel, the synaptic potential (i.e., the decrease of energy) obtained when changing its current state to each of the 3 possibilities {background, foreground, shadow}.

Let us note that, since MREM is used as an improvement of the solution provided by the mixture of Gaussians, the initial configuration of the network (i.e., the initial state vector) corresponds to the mentioned solution. Then, this solution is updated iteratively with the purpose of minimizing (3).

Suppose that neuron i is in state $v_i = k$. The expression of the synaptic potential, when updating the neuron state to k', is:

$$\begin{aligned} U_i(k, k') &= 2 \sum_{j \in \mathcal{N}_i} m_j(v_j) \left(\delta_{k, v_j} - \delta_{k', v_j}\right) + \\ &\quad + (2n_k - n) m_i(k) - (2n_{k'} - n) m_i(k') \end{aligned}$$

where n is the number of elements in the neighborhood of pixel i, and n_k and $n_{k'}$ are the number of pixels in this neighborhood in classes k and k', respectively.

Only the neuron i achieving the greatest synaptic potential $U_i(v_i, k')$ is updated to state k'. This ensures that the solution provided by the MREM network is a minimum of the energy function, that is, a solution to the segmentation problem, as proposed.

4 Experimental Results

The proposed optimization technique has been applied to a set of test sequences to show the validity of our method. These sequences have been recorded in our laboratory or downloaded from the Internet [1], with diverse kind of lighting to distinct objects in motion (people, vehicles) in order to conduct a more comprehensive study of the proposed method.

The criterion we use to determine the quality of the segmentation result is the absence of noisy or spurious objects, as well as object convexity, since these features are very important in subsequent processes, such as object tracking and behavior analysis.

Figure 1 shows the results obtained after applying MoG and our neural approach to optimize the segmentation results in two different indoor scenes. The improvement on Figures 1(e) and 1(f), with respect to the results of MoG in 1(c) and 1(d), is notable. Spurious objects are detected and removed and more convex-shaped objects are detected, thus obtaining better results.

A good object segmentation after applying the shadow detection module is assumed, for the neural optimization technique to obtain noteworthy results.

Figure 2 shows an example result on one PETS 2001 (IEEE Performance Evaluation of Tracking and Surveillance Workshops) outdoor sequence, in which the obtained objects are perfectly segmented and suitable as input of the next tracking phase.

5 Conclusions and Future Work

In this work, we have presented a hybrid method to detect moving objects in video sequences.

First, a specialized algorithm for segmenting the image is used, and then a multivalued neural model is applied to enhance the segmentation results provided by the first algorithm. This neural model allows the use of qualitative labels as neuron states, which permits a better representation of the problem.

In our case, the segmentation algorithm is based on the Mixture of Gaussian distributions model proposed by Stauffer in [16]. The initial configuration of the multivalued MREM model is set to the segmentation given by this algorithm, and the network iterates to obtain a local minimum of its energy function, measuring the quality of the segmentation.

Experimental results show that the solution proposed by our method is able to detect and correct some of the undesirable effects of common segmentation methods (such as noise, spurious objects...), giving a much clearer segmented image. This fact is beneficial for typical subsequent processes such as object tracking and behavior analysis.

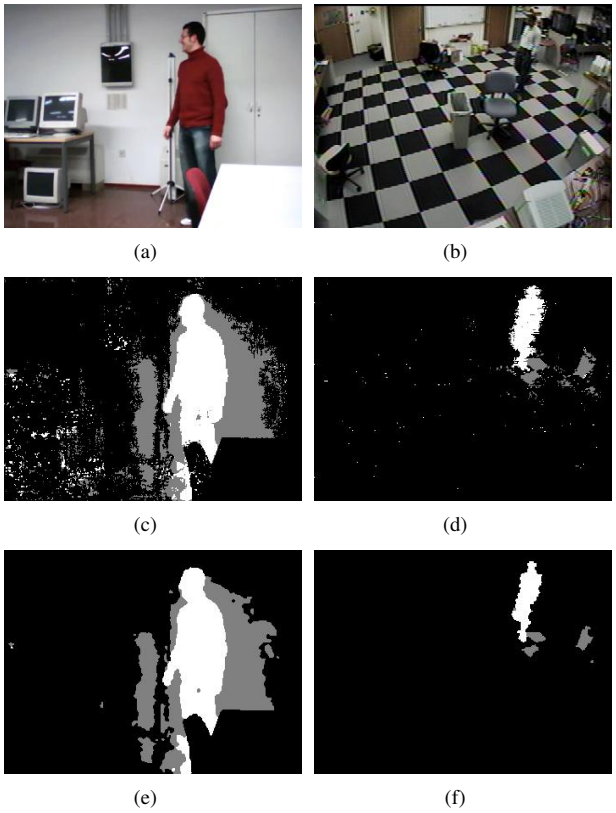

Figure 1. Results of applying our approach in several frames. (a) and (b) show the captured frames for each indoor scene in raw form; (c) and (d) show results using MoG; (e) and (f) show the final segmentation after applying our neural approach.

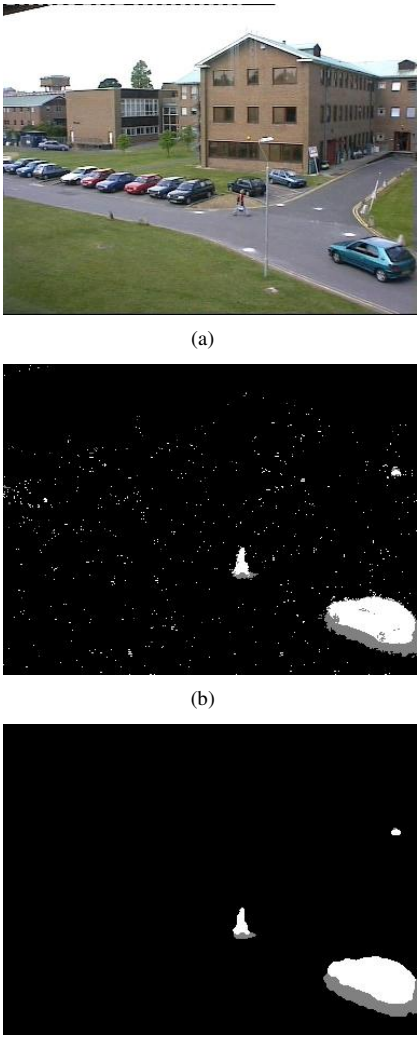

Figure 2. Results obtained in outdoor scenes. (a) is a frame obtained from the PETS01 sequence in raw form; (b) MoG method; (c) our optimization technique based on multivalued neural networks after applying MoG

In future works, we expect to improve this model by introducing a feedback process between the neural model MREM and the Mixture of Gaussians algorithm. With this improvement, the statistical model of the pixel color space can be better estimated. Thus, better segmentation results are expected.

Acknowledgments

This work is partially supported by Junta de Andalucía (Spain) under contract TIC-01615, project name Intelligent Remote Sensing Systems.

References

[1] http://www.research.ibm.com/peoplevision.

[2] C. Benedek and T. Sziranyi. Bayesian foreground and shadow detection in uncertain frame rate surveillance videos. *IEEE Transactions on Image Processing*, 17(4):608–621, 2008.

[3] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, 2003.

[4] M. H. Erdem and Y. Ozturk. A new family of multivalued networks. *Neural Networks*, 9(6):979 – 989, 1996.

[5] G. Galán-Marín, E. Mérida-Casermeiro, and D. López-Rodríguez. Improving neural networks for mechanism kinematic chain isomorphism identification. *Neural Processing Letters*, 26:133 – 143, 2007.

[6] J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. volume 79, pages 2254 – 2558, 1982.

[7] J. Hopfield and D. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:141 – 152, 1985.

[8] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proceedings of International Conference on Computer Vision*, 1999.

[9] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proceedings of the International Conference on Pattern Recognition*, 1994.

[10] B. Lo and S. Velastin. Automatic congestion detection system for underground platforms. In *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pages 158–161, 2001.

[11] D. López-Rodríguez, E. Mérida-Casermeiro, J. M. Ortiz-de Lazcano-Lobato, and G. Galán-Marín. k-pages graph drawing with multivalued neural networks. *Lecture Notes in Computer Science*, 4669:816 – 825, 2007.

[12] E. Mérida-Casermeiro. *Red Neuronal recurrente multivaluada para el reconocimiento de patrones y la optimización combinatoria*. PhD thesis, Universidad de Málaga, 2000.

[13] E. Mérida-Casermeiro and D. López-Rodríguez. Graph partitioning via recurrent multivalued neural networks. *Lecture Notes in Computer Science*, 3512:1149 – 1156, 2005.

[14] E. Mérida-Casermeiro, J. Muñoz Pérez, and R. Benítez-Rochel. A recurrent multivalued neural network for the n-queens problem. *Lecture Notes in Computer Science*, 2084:522 – 529, 2001.

[15] Y. Ozturk and H. Abut. System of associative relationships (soar). 1997.

[16] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[17] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[18] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentl. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.