

Pixel features for self-organizing map based detection of foreground objects in dynamic environments

Miguel A. Molina-Cabello¹, Ezequiel López-Rubio¹, Rafael Marcos Luque-Baena², Enrique Domínguez¹, and Esteban J. Palomo^{1,3}

¹ Department of Computer Languages and Computer Science. University of Málaga. Bulevar Louis Pasteur, 35. 29071 Málaga. Spain.

² Department of Computer Systems and Telematics Engineering. University of Extremadura. University Centre of Mérida. 06800 Mérida. Spain.

³ School of Mathematical Science and Information Technology. University of Yachay Tech. Hacienda San José s/n. San Miguel de Urucuquí. Ecuador.

Abstract. Among current foreground detection algorithms for video sequences, methods based on self-organizing maps are obtaining a greater relevance. In this work we propose a probabilistic self-organising map based model, which uses a uniform distribution to represent the foreground. A suitable set of characteristic pixel features is chosen to train the probabilistic model. Our approach has been compared to some competing methods on a test set of benchmark videos, with favorable results.

Keywords: Foreground detection, background modeling, probabilistic self-organising maps, background features

1 Introduction

Foreground object detection is a key problem in the design of computer vision systems. Algorithms to solve this problem must handle many difficulties which arise in real life videos. These inconveniences include illumination changes, shadow appearances in the foreground because of object lighting in the background or repetitive motions of background objects from the scene (waves of the sea, branches of the trees), among many others.

There are several approaches in the literature to model the background of a video sequence, employing different techniques like mixtures of Gaussians or probabilistic neural networks. In this paper we present a model based on probabilistic self-organising maps, with a suitable choice of characteristic pixel features.

The rest of the paper is structured as follows. The methodology from our proposal is described in Section 2. The experimental results are shown in Section 3. Finally we present our conclusions in Section 4.

2 Methodology

Our foreground detection system first computes the values of D features of each pixel of an incoming frame of size $NumRows \times NumCols$ pixels. The set of suitable features that we have considered is presented in [3]. After that, the feature vector $\mathbf{t} \in \mathbb{R}^D$ at pixel position $\mathbf{x} \in \{1, \dots, NumRows\} \times \{1, \dots, NumCols\}$ is provided as the input sample to a learning algorithm to adapt the parameters of a probabilistic mixture distribution with two mixture components (*Back* for the background and *Fore* for the foreground):

$$p_{\mathbf{x}}(\mathbf{t}) = \pi_{Back, \mathbf{x}} p_{\mathbf{x}}(\mathbf{t} | Back) + \pi_{Fore, \mathbf{x}} p_{\mathbf{x}}(\mathbf{t} | Fore) \quad (1)$$

The foreground values of the feature vector are modeled by the uniform distribution over the space of all possible feature vectors, so that any incoming foreground object can be represented equally well:

$$p_{\mathbf{x}}(\mathbf{t} | Fore) = U(\mathbf{t}) \quad (2)$$

$$U(\mathbf{t}) = \begin{cases} 1/Vol(\mathcal{S}) & \text{iff } \mathbf{t} \in \mathcal{S} \\ 0 & \text{iff } \mathbf{t} \notin \mathcal{S} \end{cases} \quad (3)$$

where \mathcal{S} is the support of the uniform pdf and $Vol(\mathcal{S})$ is the D -dimensional volume of \mathcal{S} . The distribution of the background values of the feature vector is represented by means of a probabilistic self-organizing map:

$$p_{\mathbf{x}}(\mathbf{t} | Back) = \frac{1}{H} \sum_{i=1}^H p_{\mathbf{x}}(\mathbf{t} | i) \quad (4)$$

where H is the number of mixture components (units) of the self-organizing map, and the prior probabilities or mixing proportions are assumed to be equal. More details about the learning algorithm for the above defined mixture are given in [2].

The Bayesian probability that the observed sample (feature vector value) \mathbf{t} is foreground is given by

$$R_{Fore, \mathbf{x}}(\mathbf{t}) = \frac{\pi_{Fore, \mathbf{x}} p_{\mathbf{x}}(\mathbf{t} | Fore)}{\pi_{Back, \mathbf{x}} p_{\mathbf{x}}(\mathbf{t} | Back) + \pi_{Fore, \mathbf{x}} p_{\mathbf{x}}(\mathbf{t} | Fore)} \quad (5)$$

However, $R_{Fore, \mathbf{x}}(\mathbf{t})$ is prone to noise due to isolated pixels that change their features randomly. The Pearson correlations $\rho_{\mathbf{x}, \mathbf{y}}$ allow us to obtain a noise-reduced version of $R_{Fore, \mathbf{x}}(\mathbf{t})$ by combining it with the information from the 8-neighbours \mathbf{y} of \mathbf{x} :

$$\tilde{R}_{Fore, \mathbf{x}}(\mathbf{t}) = \text{trunc} \left(\frac{1}{9} \sum_{\mathbf{y} \in Neigh(\mathbf{x})} \rho_{\mathbf{x}, \mathbf{y}} R_{Fore, \mathbf{y}}(\mathbf{t}) \right) \quad (6)$$

where $Neigh(\mathbf{x})$ contains the pixel \mathbf{x} and its 8-neighbours \mathbf{y} .

Table 1. Summary of the main model characteristics for each compared method.

Name	Model characteristics
WrenGA	One Gaussian distribution
GrimsonGMM	K Gaussian distributions
MaddalenaSOBS	Artificial neural networks
FSOM	Uniform distribution and probabilistic self-organizing map

3 Experimental results

In this section we present the foreground detection performance of our method and a comparison with other algorithms of the state-of-art. Software and hardware used in our experiments are shown in Subsection 3.1. We detail the tested sequences in Subsection 3.2. The set of parameters by each method are specified in Subsection 3.3. Finally the qualitative and quantitative results are reported in Subsection 3.4.

3.1 Methods

Our method called FFSOM is based on the object detection method FSOM [2], which was previously developed by our research group and it is included in the comparisons. The code of this method can be downloaded for free⁴.

The FFSOM and FSOM methods have been implemented in Matlab, using MEX files written in C++ for those quite time-demanding parts.

Additionally we have selected some reference methods of the literature. The first we used is the algorithm we note as WrenGA [7], which is the oldest and it features a single Gaussian. Other Gaussians approach method is the one we name GrimsonGMM [5], that uses two Mixture of Gaussians. Finally we have chosen an artificial neural networks approach method noted MaddalenaSOBS [4]. The main characteristics of all selected methods are shown in Table 1.

The implementation of these tested methods have been taken from the BGS libray version 1.3.0, which is accessible from its website⁵.

Since our FFSOM and FSOM methods include a post-processing and the MaddalenaSOBS method has an implicit post-processing, we have added post-processing to all the other methods so as to make the comparisons as fair as possible.

The experiments reported in this paper have been carried out on a 64-bit Personal Computer with an eight-core Intel i7 3.60 GHz CPU, 32 GB RAM and standard hardware. The implementation of our method does not use any GPU resources, so it does not require any specific graphics hardware.

⁴ <http://www.lcc.uma.es/%7Eezeqlr/fsom/fsom.html>

⁵ <https://github.com/andrewsobral/bgslibrary>

Table 2. Considered parameter values for the competing methods, forming the set of experimental configurations.

Method	Parameters
FFSOM	Features, $F = \{[1\ 19\ 20]\}$ Step size, $\alpha = \{0.01\}$ Number of neurons, $N = \{12\}$
FSOM	Step size, $\alpha = \{0.01\}$ Number of neurons, $N = \{12\}$
GrimsonGMM	Threshold, $T = \{12\}$ Learning rate, $\alpha = \{0.0025\}$ Number of Gaussians in the mixture model, $K = \{3\}$
MaddalenaSOBS	Sensitivity, $s_1 = \{75\}$ Training sensitivity, $s_0 = \{245\}$ Learning rate, $\alpha_1 = \{75\}$ Training step, $N = \{100\}$
WrenGA	Threshold, $T = \{12\}$ Learning rate, $\alpha = \{0.005\}$

3.2 Sequences

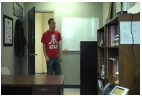























A set of videos have been selected from the 2014 dataset of the ChangeDetection.net website⁶. The sequences have been chosen are two videos from the Baseline category and other two from the Low Framerate category. The first one contains simple videos and the other one is composed by sequences with low frame rate. The video *Office* presents a room and a person who appears, he stays with low movements and then he goes out (360x240 pixels and 2050 frames), and *PETS2006* shows a train station with people moving on in (720x576 pixels and 1200 frames). This two videos are from the Baseline category. On the other hand, the two sequences selected from the Low Framerate category are *TramCrossroad*, a crossroad with cars driving for different ways (640x350 pixels and 900 frames); and *Turnpike* (320x240 pixels and 1500 frames), a highway with cars moving from left to right and vice versa.

3.3 Parameter selection

We have defined a set of fixed values for the parameters of the methods to make the comparisons. The tuned values of each method are selected from the author’s recommendations and they are shown in Table 2.

⁶ <http://changedetection.net/>

Fig. 1. Qualitative results for some benchmark scenes. From left to right: frame 1638 from Office (a), frame 956 from PETS2006 (b), frame 420 from TramCrossroad (c) and frame 958 from Turnpike (d) respectively. The first and second rows correspond to the original video frame and the ground truth. The remaining rows are the results given by the compared methods.

	(a)	(b)	(c)	(d)
Frame				
GT				
FFSOM				
FSOM				
GrimsonGMM				
MaddalenaSOBS				

3.4 Results

On the one hand, from a qualitative point of view, in most cases the produced results by all compared methods are very similar, as it can be shown in Figure 1.

On the other hand there are other scenes from the tested sequences for all the methods whose segmented images are obtained with noise and this promotes worst quantitative results. Furthermore there are other presented problems like camouflage (pixels from foreground and background are very similar) or sudden lighting changes in the scene.

The goodness of a method and the comparison with others can be evaluated with different quantitative performance measures. One of them we have selected is the called *spatial accuracy*, which it has been used for the comparisons in other

Table 3. Accuracy results (higher is better). Each column corresponds to a video and the rows indicate the methods. Each cell shows the mean and standard deviation of the accuracy over all tested configurations. Best results are highlighted in **bold**.

Method	Office	PETS2006	TramCrossroad	Turnpike
FFSOM	0.569±0.148	0.679±0.077	0.073±0.133	0.317±0.321
FSOM	0.535±0.147	0.658±0.082	0.071±0.129	0.298±0.302
GrimsonGMM	0.285±0.141	0.501±0.157	0.069±0.124	0.293±0.297
MaddalenaSOBS	0.701±0.118	0.638±0.086	0.053±0.101	0.299±0.302
WrenGA	0.350±0.154	0.448±0.154	0.067±0.122	0.262±0.267

papers[1, 6], and it is defined as follows:

$$AC = \frac{\text{card}(A \cap B)}{\text{card}(A \cup B)} \quad (7)$$

where 'card' stands for the number of elements of a set, A is the set of all pixels which belong to the foreground, and B is the set of all pixels which are classified as foreground by the analyzed method.

Furthermore, the F-measure is also employed, which is a proportion of the precision (PR) and recall (RC) metrics and it is defined as follows:

$$F - \text{measure} = 2 * \frac{PR * RC}{PR + RC} \quad (8)$$

The average accuracy with its standard deviation for the best configuration for each sequence are shown in Table 3. Furthermore some different results are presented in Figures 2, 3 and 4.

FFSOM presents the best performance of all tested methods in three of the four analyzed videos. In addition, FFSOM obtained better results than FSOM.

Other significant aspect is the low accuracy presented in the TramCrossroad sequence by all methods. This is motivated because the ground truth of the different images from the video are not complete as we can see in Figure 1, row GT, column (c).

4 Conclusions

We have proposed a new method for the background modeling and foreground object detection using a set of pixel features different from the standard RGB values. The method has been compared with other state-of-art algorithms employing several benchmark videos. The reported results are satisfactory, with our method achieving the best performance in the majority of the tests.

Acknowledgments

This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grant TIN2014-53465-R, project name Video surveillance by active search of anomalous events. It is also partially supported by the Autonomous Government of Andalusia (Spain) under projects TIC-6213, project name Development of Self-Organizing Neural Networks for Information Technologies; and TIC-657, project name Self-organizing systems and robust estimators for video surveillance. Finally, it is partially supported by the Autonomous Government of Extremadura (Spain) under the project IB13113. All of them include funds from the European Regional Development Fund (ERDF). The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga.

References

1. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing* 13(11), 1459–1472 (Nov 2004)
2. López-Rubio, E., Luque-Baena, R.M., Dominguez, E.: Foreground detection in video sequences with probabilistic self-organizing maps. *International Journal of Neural Systems* 21(03), 225–246 (2011)
3. López-Rubio, F.J., López-Rubio, E.: Features for stochastic approximation based foreground detection. *Computer Vision and Image Understanding* 133, 30 – 50 (2015)
4. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing* 17(7), 1168–1177 (2008)
5. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: *Proc. IEEE Intl. Conf. on Computer Vision and Pattern Recognition*. pp. 246–252. Fort Collins, Colorado (1999)
6. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: *IEEE International Conference on Computer Vision, ICCV*. pp. 255–261. Kerkyra, Greece (1999)
7. Wren, C., Azarbayejani, A., Darrell, T., Pentl, A.: Pfunder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7), 780–785 (1997)

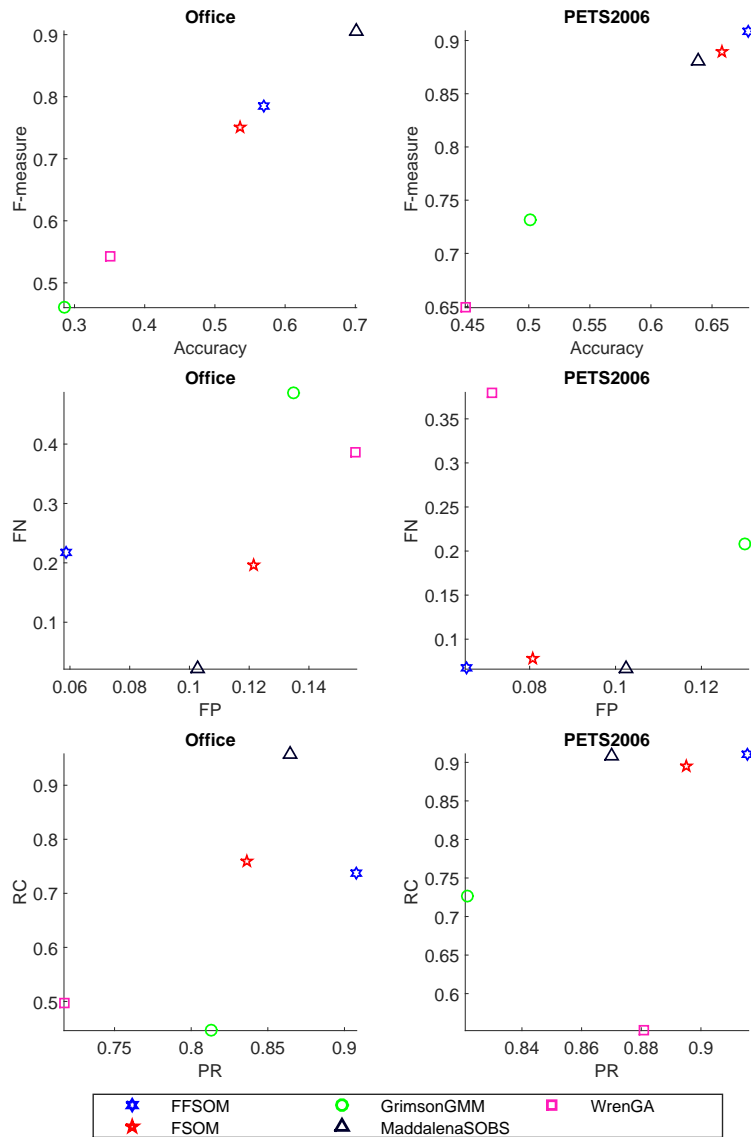


Fig. 2. Accuracy versus F-measure, False positives versus False negatives, and Precision versus Recall for each method. First column shows the Office sequence and the second column corresponds to the PETS2006 video.

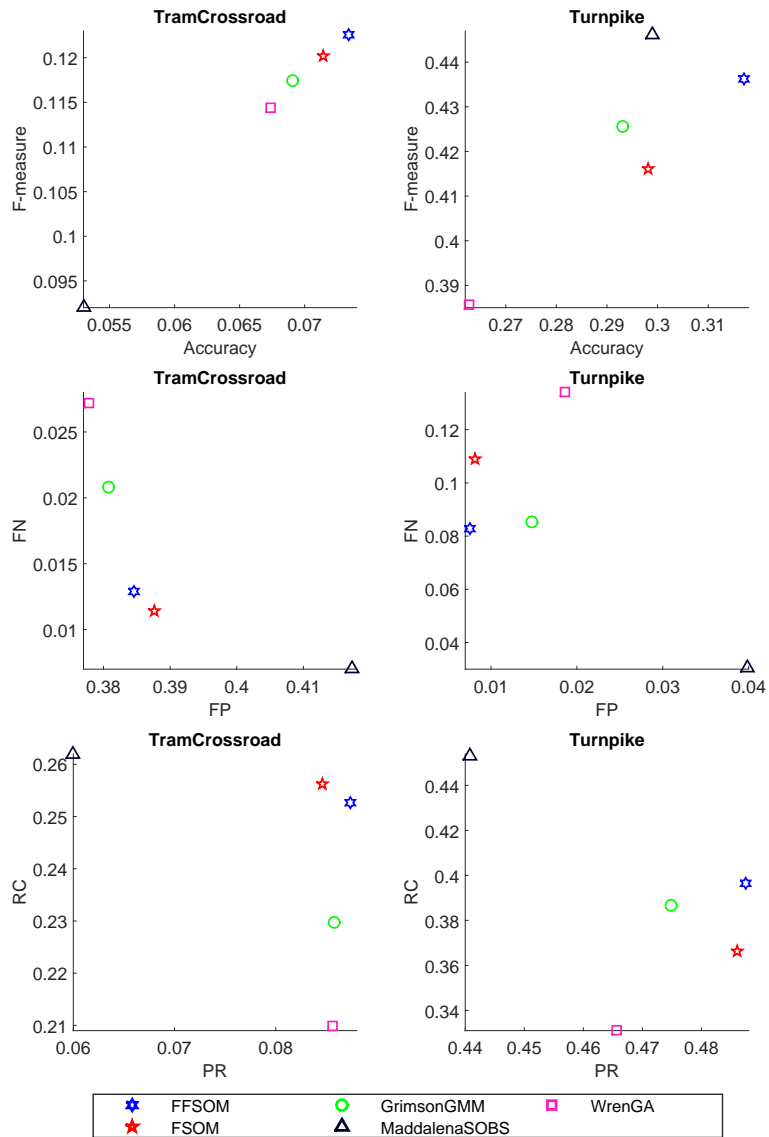


Fig. 3. Accuracy versus F-measure, False positives versus False negatives, and Precision versus Recall for each method. First column shows the TramCrossroad sequence and the second column corresponds to the Turnpike video.

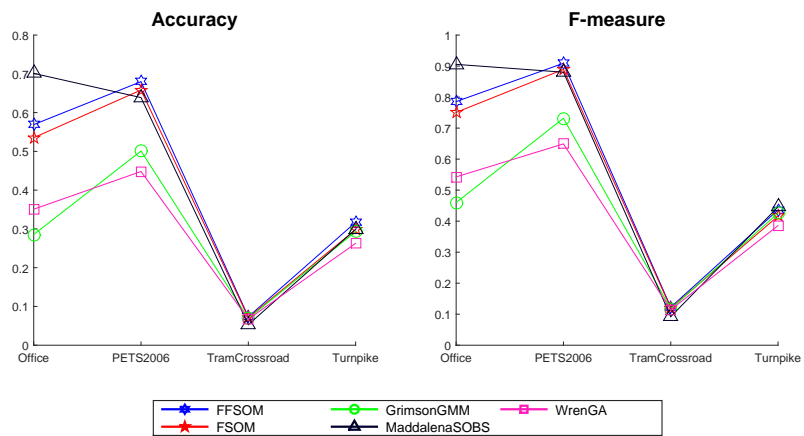


Fig. 4. Accuracy and F-measure for each method in each video. Please note that the values of each method are connected between them with lines to appreciate which method is better in each video, but this does not mean that the videos are related.