2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Color Space Selection for Self-Organizing Map Based Foreground Detection in Video Sequences

Francisco Javier López-Rubio, Ezequiel López-Rubio, Rafael Marcos Luque-Baena,
Enrique Domínguez and Esteban J. Palomo

***Abstract*—The selection of the best color space is a fundamental task in detecting foreground objects on scenes. In many situations, especially on dynamic backgrounds, neither grayscale nor RGB color spaces represent the best solution to detect foreground objects. Other standard color spaces, such as YCbCr or HSV, have been proposed for background modeling in the literature; although the best results have been achieved using diverse color spaces according to the application, scene, algorithm, etc. In this work, a color space and color component weighting selection process is proposed to detect foreground objects in video sequences using self-organizing maps. Experimental results are also provided using well known benchmark videos.**

I. Introduction

BACKGROUND modeling constitutes a key task of many computer vision applications, such as video surveillance systems. When analyzing a video sequence, foreground objects are extracted to be used in subsequent classifying or recognizing phases. The best known approach for detecting these objects of interest is the background subtraction process, where a static background model is subtracted from the current image. This simple method presents a poor performance against illumination changes, noise or non-static backgrounds, increasing the number of false positives in foreground objects detection. In order to overcome such limitations, a moving-objects detection method based on Cauchy distribution for video surveillance systems was proposed in [1]. In this method, the distribution of the intensity ratios between corresponding pixels of two background images obeys a Cauchy distribution. Based on the change detection, the intensity, hue and saturation in the YCbCr color space are employed to recognize and eliminate shadows and reflections in video sequences.

The election of the color space is essential for the good performance of a foreground detection method. In [2], the HSV color space is used for background modeling, which is combined with moving object segmentation based on fuzzy clustering to extract objects of interest from frames. The accurate description of the HSV color space is able to restore the background and then the moving object segmentation is used to distinguish the moving area and noise area. The election of an appropriate color space is further investigated in [3], where a hybrid color space constituted by the three significant color components is determined. This approach is used for color image segmentation in a soccer video with a non-static background. Another approach [4] uses the YCbCr color space to introduce a vehicle shadow segmentation algorithm for moving vehicle detection in a traffic monitoring system. This approach proposes a background subtraction method based on binary discrete wavelet transforms (BDWT). The BDWT is used together with the shadow segmentation algorithm to obtain the motion area in the Y component and then segments the shadow in the YCbCr color space. In [5], a Hybrid Cone-Cylinder Codebook (HC3) model is introduced, which combines an adaptive background model with HSV-color space shadow suppression. The background subtraction problem when there is a non-stationary background is also addressed in [6], where a Gaussian Mixture Model is used for background modeling. Moreover, a different color space named Lab2000HL is used in addition to usual color spaces, which has a linear hue band.

In an earlier work [7] a background model based on probabilistic self-organizing maps was proposed. A fundamental limitation of this approach is that it considers spherical covariance matrices, which means that all the input dimensions are weighted equally. Moreover, only the standard RGB color space was considered. Here we aim to address these issues by selecting the most appropriate color space and color component weighting for a given scene.

The structure of this paper is as follows. First the basic framework is reviewed (Section II). Then the color space and component weighting selection are discussed (Section III). Finally, Sections IV and V are devoted to experimental results and conclusions, respectively.

II. Background model

In this section the basic foreground detection system is outlined. First of all the probabilistic background model is described, which is based on a self-organizing map (Subsection II-A). After that, an online learning process based on stochastic approximation is explained (Section II-B). More details can be found in [7].

Francisco Javier López-Rubio, Ezequiel López-Rubio, Enrique Domínguez and Esteban J. Palomo are with the Department of Computer Languages and Computer Science, University of Málaga, 29071 Málaga, Spain, e-mails: xavierprof@hotmail.com, {ezeqlr,enriqued,ejpalomo}@lcc.uma.es

Rafael Marcos Luque-Baena is with the Department of Computer Systems and Telematics Engineering, University of Extremadura, 06800 Mérida, Spain, e-mail: rmluque@unex.es

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

A. Model definition

The model receives the incoming video frames and processes their pixels as training samples. It it aimed to build a probabilistic representation of the background of the scene, which is used to determine which pixels belong to the foreground at each time step. As we will see, many color spaces can be used [8], but in all cases the input space dimension is $D = 3$, i.e. tristimulus color values are considered. The probability distribution of the pixel color value $\mathbf{t}$ at pixel position $\mathbf{x}$ is modeled by a mixture with two components, namely $Back$ for the background and $Fore$ for the foreground. The associated probability density function is given by:

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

$$\pi_{Back,\mathbf{x}} + \pi_{Fore,\mathbf{x}} = 1 \quad (2)$$

It must be highlighted that for each pixel location $\mathbf{x}$ a probabilistic mixture is defined in (1). This way the model is able to adapt to the specific characteristics of every pixel of the scene. In general terms it can be assumed that foreground objects can have any color. This calls for a uniform distribution over the color space to model the foreground:

$$p_{\mathbf{x}}(\mathbf{t} \mid Fore) = U(\mathbf{t}) \quad (3)$$

$$U(\mathbf{t}) = \begin{cases} 1/Vol(\mathcal{S}) & \text{iff } \mathbf{t} \in \mathcal{S} \\ 0 & \text{iff } \mathbf{t} \notin \mathcal{S} \end{cases} \quad (4)$$

where $\mathcal{S}$ is the overall color space and $Vol(\mathcal{S})$ is the three dimensional volume of $\mathcal{S}$. This way to model the foreground ensures that all incoming objects are treated the same way, irrespective of their color.

The distribution of the background color values at a certain position $\mathbf{x}$ depends on the features of the scene. For example, waving trees and other dynamic background objects lead to background distributions which are multimodal. A probabilistic self-organizing map can cope with this kind of distribution, since each neuron can specialize on one cluster of the input dataset:

$$p_{\mathbf{x}}(\mathbf{t} \mid Back) = \frac{1}{H} \sum_{i=1}^{H} p_{\mathbf{x}}(\mathbf{t} \mid i) \quad (5)$$

where H is the number of mixture components (neurons) of the self-organizing map, and the prior probabilities or mixing proportions are assumed to be equal. Now it is necessary to define a *topological distance* $d(i,j)$ for each pair of neurons (i,j) of the map. The standard choice has been used here: a rectangular grid to place the units, together with the Euclidean topological distance:

$$d(i,j) = \|\mathbf{r}_i - \mathbf{r}_j\| \quad (6)$$

where $\mathbf{r}_i$ and $\mathbf{r}_j$ are the positions of units i and j in the rectangular grid, respectively.

The computational load of the probabilistic model should be as small as possible, since there is one self-organizing map for each position $\mathbf{x}$ in the video frame. The simplest option is to consider a spherical Gaussian probability density for each mixture component $i \in \{1, ..., H\}$ of the map [9], [10], [11]:

$$p_{\mathbf{x}}(\mathbf{t} \mid i) = (2\pi)^{-D/2} \left(\sigma^2_{i,\mathbf{x}}\right)^{-D/2} \exp\left(-\frac{1}{2\sigma^2_{i,\mathbf{x}}} \left(\mathbf{t} - \boldsymbol{\mu}_{i,\mathbf{x}}\right) \left(\mathbf{t} - \boldsymbol{\mu}_{i,\mathbf{x}}\right)^T\right) \quad (7)$$

where $\boldsymbol{\mu}_{i,\mathbf{x}}$ and $\sigma^2_{i,\mathbf{x}}$ are the mean vector and the variance of mixture component *i*, respectively:

$$\boldsymbol{\mu}_{i,\mathbf{x}} = E[\mathbf{t} \mid i, \mathbf{x}] \quad (8)$$

$$\sigma^2_{i,\mathbf{x}} = E\left[\frac{1}{D} \left(\mathbf{t} - \boldsymbol{\mu}_{i,\mathbf{x}}\right)^T \left(\mathbf{t} - \boldsymbol{\mu}_{i,\mathbf{x}}\right) \mid i, \mathbf{x}\right] \quad (9)$$

In order to decide whether an observed pixel belongs to the background, a Bayesian classification procedure is carried out. The probability that the observed data (pixel color value) $\mathbf{t}$ is background is given by

$$P_{Back,\mathbf{x}}(\mathbf{t}) = \frac{\pi_{Back,\mathbf{x}} p_{\mathbf{x}}(\mathbf{t} \mid Back)}{\pi_{Back,\mathbf{x}} p_{\mathbf{x}}(\mathbf{t} \mid Back) + \pi_{Fore,\mathbf{x}} p_{\mathbf{x}}(\mathbf{t} \mid Fore)} \quad (10)$$

and the corresponding probability of the foreground is the complementary event:

$$P_{Fore,\mathbf{x}}(\mathbf{t}) = 1 - P_{Back,\mathbf{x}}(\mathbf{t}) \quad (11)$$

There are many undesirable effects that introduce noise in $P_{Back,\mathbf{x}}(\mathbf{t})$ and $P_{Fore,\mathbf{x}}(\mathbf{t})$. For example, camouflage effects (foreground objects whose color is similar to that of the background), camera imperfections and video compression artifacts. In order to alleviate this problem, the information from the 8-neighbors of a given pixel $\mathbf{x}$ can be used to reduce the noise. A simple approach would be to weight all neighbors equally (low pass filter), but this would neglect the fact that many neighboring pixels are not related. For example, this can happen on the bank of a river: the pixels outside the water (where no waves occur) are almost independent from those inside the river (where the water current changes the surface) despite of their proximity. A principled way to measure the correlation of pairs of pixels is Pearson's correlation [12] between the random variables $P_{Fore,\mathbf{x}}$ and $P_{Fore,\mathbf{y}}$ corresponding to each pair of 8-neighboring pixels $\mathbf{x}$ and $\mathbf{y}$:

$$\rho_{\mathbf{x},\mathbf{y}} = \frac{\phi_{\mathbf{x},\mathbf{y}}}{\sqrt{\nu_{\mathbf{x}}}\sqrt{\nu_{\mathbf{y}}}} \quad (12)$$

$$\phi_{\mathbf{x},\mathbf{y}} = \mathrm{cov}(P_{Fore,\mathbf{x}}, P_{Fore,\mathbf{y}}) = E\left[\left(P_{Fore,\mathbf{x}} - E[P_{Fore,\mathbf{x}}]\right)\left(P_{Fore,\mathbf{y}} - E[P_{Fore,\mathbf{y}}]\right)\right] \quad (13)$$

$$\nu_{\mathbf{x}} = \mathrm{var}(P_{Fore,\mathbf{x}}) = E\left[\left(P_{Fore,\mathbf{x}} - E[P_{Fore,\mathbf{x}}]\right)^2\right] \quad (14)$$

$$\nu_{\mathbf{y}} = \mathrm{var}\left(P_{Fore,\mathbf{y}}\right) = E\left[\left(P_{Fore,\mathbf{y}} - E\left[P_{Fore,\mathbf{y}}\right]\right)^2\right] \quad (15)$$

where we have:

$$E\left[P_{Fore,\mathbf{x}}\right] = \pi_{Fore,\mathbf{x}} \quad (16)$$

$$E\left[P_{Fore,\mathbf{y}}\right] = \pi_{Fore,\mathbf{y}} \quad (17)$$

Pearson's correlation is bounded and symmetric, i.e.

$$\rho_{\mathbf{x},\mathbf{y}} \in [-1, 1] \quad (18)$$

$$\rho_{\mathbf{x},\mathbf{y}} = \rho_{\mathbf{y},\mathbf{x}} \quad (19)$$

where the last equation can be used to save one half of the computations.

If two neighboring pixels are usually assigned to the same class, i.e. either both are background or both are foreground, then the correlation between them $\rho_{\mathbf{x},\mathbf{y}}$ is large and positive. On the other hand, if both pixels are independent, then we have $\rho_{\mathbf{x},\mathbf{y}} = 0$. A negative correlation corresponds to a pair of pixels which are usually assigned to opposite classes. This is rather unlikely, but there are some cases where small negative values are obtained due to noise.

The noise in $P_{Fore,\mathbf{x}}\left(\mathbf{t}\right)$ can be reduced with the help of the correlations $\rho_{\mathbf{x},\mathbf{y}}$, so that the 8-neighbors of $\mathbf{x}$ with the highest positive correlations are given more importance:

$$\widetilde{P}_{Fore,\mathbf{x}}\left(\mathbf{t}\right) = \mathrm{trunc}\left(\frac{1}{9}\sum_{\mathbf{y}\in Neigh(\mathbf{x})}\rho_{\mathbf{x},\mathbf{y}}P_{Fore,\mathbf{y}}\left(\mathbf{t}\right)\right) \quad (20)$$

where $Neigh\left(\mathbf{x}\right)$ contains the pixel $\mathbf{x}$ and its 8-neighbours, and

$$\rho_{\mathbf{x},\mathbf{x}} = 1 \quad (21)$$

$$\mathrm{trunc}\left(z\right) = \begin{cases} z & \text{iff } z \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

$$\widetilde{P}_{Fore,\mathbf{x}}\left(\mathbf{t}\right) \in [0, 1] \quad (23)$$

The role of the trunc function in (20) is to deactivate the parameter learning by setting $\widetilde{P}_{Fore,\mathbf{x}}\left(\mathbf{t}\right) = 0$ when a spike of noisy negative correlations occurs. It must be pointed out that the final result of this procedure is postprocessed by filling holes of size one pixel; then the objects with less than 10 pixels in size are removed.

In the following subsection a possible way to train the above model is considered. Stochastic approximation has been chosen for this purpose [13]; it has been used before to develop online mode learning procedures for probabilistic self-organizing maps [14], [15]. One of its main advantages is that its associated computational complexity $O\left(HD\right)$ is quite low, which is of paramount importance for background models based on self-organizing maps [16].

B. Learning

The parameter update procedure is a obtained by application of Robbins-Monro stochastic approximation algorithm [13]. Let us remember that $\widetilde{P}_{Fore,\mathbf{x}}\left(\mathbf{t}_{n,\mathbf{x}}\right)$ is an enhanced (denoised) estimation of the a posteriori probability that the input sample $\mathbf{t}_{n,\mathbf{x}}$ at pixel position $\mathbf{x}$ and time instant n belongs to the foreground:

$$P_{\mathbf{x}}\left(Fore \mid \mathbf{t}_{n,\mathbf{x}}\right) = \widetilde{P}_{Fore,\mathbf{x}}\left(\mathbf{t}_{n,\mathbf{x}}\right) \quad (24)$$

$$P_{\mathbf{x}}\left(Back \mid \mathbf{t}_{n,\mathbf{x}}\right) = \widetilde{P}_{Back,\mathbf{x}}\left(\mathbf{t}_{n,\mathbf{x}}\right) = 1 - P_{\mathbf{x}}\left(Fore \mid \mathbf{t}_{n,\mathbf{x}}\right) \quad (25)$$

The system learns in online mode. That is, at time instant n a new input frame (image) is acquired and fed to the system, and the parameters for every pixel location $\mathbf{x}$ are modified according to the pixel color $\mathbf{t}_{n,\mathbf{x}}$ at that location. The application of Robbins-Monro algorithm yields the following update equations:

$$\pi_{Fore,\mathbf{x}}\left(n\right) = \left(1 - \epsilon\left(n\right)\right)\pi_{Fore,\mathbf{x}}\left(n-1\right) + \epsilon\left(n\right)\widetilde{P}_{Fore,\mathbf{x}}\left(\mathbf{t}_{n,\mathbf{x}}\right) \quad (26)$$

$$\pi_{Back,\mathbf{x}}\left(n\right) = 1 - \pi_{Fore,\mathbf{x}}\left(n\right) \quad (27)$$

$$\nu_{\mathbf{x}}\left(n\right) = \left(1 - \epsilon\left(n\right)\right)\nu_{\mathbf{x}}\left(n-1\right) + \epsilon\left(n\right)\left(\widetilde{P}_{Fore,\mathbf{x}}\left(\mathbf{t}_{n,\mathbf{x}}\right) - \pi_{Fore,\mathbf{x}}\left(n\right)\right)^2 \quad (28)$$

$$\phi_{\mathbf{x},\mathbf{y}}\left(n\right) = \left(1 - \epsilon\left(n\right)\right)\phi_{\mathbf{x},\mathbf{y}}\left(n-1\right) + \epsilon\left(n\right)\left(\widetilde{P}_{Fore,\mathbf{x}}\left(\mathbf{t}_{n,\mathbf{x}}\right) - \pi_{Fore,\mathbf{x}}\left(n\right)\right)\left(\widetilde{P}_{Fore,\mathbf{y}}\left(\mathbf{t}_{n,\mathbf{y}}\right) - \pi_{Fore,\mathbf{y}}\left(n\right)\right) \quad (29)$$

$$\rho_{\mathbf{x},\mathbf{y}}\left(n\right) = \left(1 - \epsilon\left(n\right)\right)\rho_{\mathbf{x},\mathbf{y}}\left(n-1\right) + \epsilon\left(n\right)\frac{\phi_{\mathbf{x},\mathbf{y}}\left(n\right)}{\sqrt{\nu_{\mathbf{x}}\left(n\right)}\sqrt{\nu_{\mathbf{y}}\left(n\right)}} \quad (30)$$

$$\boldsymbol{\mu}_{i,\mathbf{x}}\left(n\right) = \left(1 - \xi_{\mathbf{x}}\left(i,n\right)\right)\boldsymbol{\mu}_{i,\mathbf{x}}\left(n-1\right) + \xi_{\mathbf{x}}\left(i,n\right)\mathbf{t}_{n,\mathbf{x}} \quad (31)$$

$$\sigma^2_{i,\mathbf{x}}\left(n\right) = \left(1 - \xi_{\mathbf{x}}\left(i,n\right)\right)\sigma^2_{i,\mathbf{x}}\left(n-1\right) + \xi_{\mathbf{x}}\left(i,n\right)\frac{1}{D}\left(\mathbf{t}_{n,\mathbf{x}} - \boldsymbol{\mu}_{i,\mathbf{x}}\left(n\right)\right)^T\left(\mathbf{t}_{n,\mathbf{x}} - \boldsymbol{\mu}_{i,\mathbf{x}}\left(n\right)\right) \quad (32)$$

$$\xi_{\mathbf{x}}\left(i,n\right) = \epsilon\left(n\right)\frac{\Lambda\left(i, Winner\left(\mathbf{x},n\right)\right)}{\pi_{Back,\mathbf{x}}\left(n\right)}\widetilde{P}_{Back,\mathbf{x}}\left(\mathbf{t}_{n,\mathbf{x}}\right) \quad (33)$$

Please note that a Gaussian *neighborhood function* Λ (not to be confused with the probability density function $p_{\mathbf{x}}\left(\mathbf{t} \mid i\right)$ of the mixture components) is used in (33). The neighborhood function varies with the time step *n* according

to a decaying *neighborhood radius* $\Delta(n)$ and the topological distance $d(i,j)$ to the winner:

$$\Lambda(i, Winner(\mathbf{x}, n)) = \exp\left(-\left(\frac{d(i, Winner(\mathbf{x}, n))}{\Delta(n)}\right)^2\right) \quad (34)$$

$$\Delta(n+1) \leq \Delta(n) \quad (35)$$

It must be emphasized that we use $\widetilde{R}_{Fore,\mathbf{x}}$ and $\widetilde{R}_{Back,\mathbf{x}}$ instead of $R_{Fore,\mathbf{x}}$ and $R_{Back,\mathbf{x}}$ in equations (26), (28), (29) and (33) because the two former values are denoised versions of the latter. This way the background/foreground probabilities of the neighboring pixels are fed back to the learning procedure. The overall effect is that the pixels which lie in the same region of the image cooperate with each other, and more coherent results are produced.

As in any other application, the probabilistic mixture learning method we have just outlined can fall into a sub-optimal solution, i.e. one which does not represent the input distribution faithfully. In order to tackle both problems at a time two thresholds are defined, σ^2_{max} and σ^2_{min}, and it is required that each unit i satisfies the following condition:

$$\sigma^2_{min} \leq \sigma^2_{i,\mathbf{x}} \leq \sigma^2_{max} \quad (36)$$

This is enforced by setting $\sigma^2_{i,\mathbf{x}}$ to σ^2_{max} or σ^2_{min} every time that the update equation (32) produces a new value that does not fulfill (36).

Until now, the nature of the three dimensional inputs $\mathbf{t}$ has not been specified. In the following section, the role of the color space and the weighting of the color components is studied, so that a proper choice of the input information to the background model is carried out.

III. Color spaces and color component weighting

In most commercial video cameras the pixels are given in the RGB color space with 8-bit precision values. If the raw color information from a camera of this kind was used, then we would have:

$$\mathcal{S}_{RGB} = \{(t_R, t_G, t_B) \,|\, t_R, t_G, t_B \in [0, 255]\} \quad (37)$$

$$Vol(\mathcal{S}_{RGB}) = 255^3 \quad (38)$$

However, it is well known that the RGB color space does not reflect the true similarities among colors [6]. Moreover, depending on the scene one color component could be more informative than the others, so it should be given more importance.

Let us consider a general color space $\mathcal{S}$, with colors $\mathbf{t} = (t_1, t_2, t_3) \in \mathcal{S}$. The difference between two colors $\mathbf{t}_A, \mathbf{t}_B \in \mathcal{S}$, which is used in (7) to adapt the probabilistic model, is the Euclidean distance in S:

$$\delta(\mathbf{t}_A, \mathbf{t}_B) = \|\mathbf{t}_A - \mathbf{t}_B\|^2 \quad (39)$$

Under this basic setting, the three color components have the same importance in the distance computation. Now, it is possible that a different weighting of the components yields a better performance:

$$\mathbf{t}' = (\alpha t_1, \beta t_2, \gamma t_3) \quad (40)$$

$$\delta(\mathbf{t}'_A, \mathbf{t}'_B) = \alpha(t_{A1} - t_{B1})^2 + \beta(t_{A2} - t_{B2})^2 + \gamma(t_{A3} - t_{B3})^2 \quad (41)$$

where the scale factors are non negative:

$$\alpha, \beta, \gamma \geq 0 \quad (42)$$

Next, we must take into account that spherical covariance Gaussians are equivariant with respect to homogeneous scalings, i.e. if all the dimensions are scaled by the same factor, then the learned model parameters are scaled accordingly [17]. This implies that there are only two degrees of freedom in the choice of the scale factors, since we may normalize the transformed components to have:

$$\alpha + \beta + \gamma = 1 \quad (43)$$

Please note that (42) and (43) are equivalent to:

$$\alpha, \beta, \gamma \in [0, 1] \quad (44)$$

$$\gamma = 1 - \alpha - \beta \quad (45)$$

Consequently an optimization process can be carried out on α and β according to (44) and (45) in order to choose the best scaling factors for the background modeling task at hand.

IV. Experimental results

In this section we explain what tests have been conducted to justify our conclusions. First we list the color spaces used, then we describe the sequences, tested parameters and the performance measures and finally, we report the results both qualitatively and quantitatively.

A. Color spaces

We have chosen four different color spaces besides the RGB space. The first is the CIELAB space, named as Lab, second was tested Luv, both established in 1976 by the Commission Internationale de l'éclairage (CIE). Thirdly we tested the space HSV (Hue, Saturation, Value) and finally the YCbCr space.

B. Sequences

In order to make tests as fair as possible, it was necessary to use a set of sequences that represent real situations such as indoor and outdoor environments. For this reason we employed a total of 11 sequences. All tested sequences have been used in other studies and are publicly accessible. In some cases it has been necessary to segment the sequences manually in order to perform the quantitative study. This is not the case of Video2 and Video4 created by the International Conference VSSN'06[1], because in these sequences there are artificially inserted 3D objects and therefore it is easy to determine which points correspond to the foreground.

A set of complex sequences developed by Li et al. and available in his website[2] has been used: Campus, Meeting Room, Subway Station, Water Surface and Lobby; these sequences suffer typical kinds of artifacts that complicate the segmentation, such as situations with camouflage, cast shadows and illumination changes.

We have also selected other videos used in the literature, such as a video from CAVIAR dataset[3] named Corridor (also called OneShopOneWait1cor) which presents a busy corridor that has been difficult to segment manually.

Fountain and LevelCrossing are other examples of outdoor videos. Finally we add a sequence with a significant amount of abrupt illumination changes called LightSwitch.

C. Parameter selection

To weight each component of the input videos we test the following values: $\alpha, \beta, \gamma = \{0.0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0\}$ we also considered the case of $\alpha = \beta = \gamma = 0.33$, so in total there are a set of 22 different configurations. On the other hand it is also necessary to adjust SOM model to each color space, mainly the learning rate l, the step size ε_0 and the number of neurons H; tests have proven that the values shown in Table I get good results. The values of l and ε_0 control the amount of learning (higher means faster adaptation to the input), while H controls the size of the background model.

Table I
SET OF PARAMETERS USED BY THE SOM MODEL IN EACH COLOR SPACE

Color Space	l	ε_0	H
RGB	0.050	0.050	18
Lab	0.001	0.010	18
Luv	0.005	0.001	6
HSV	0.010	0.050	6
YCbCr	0.005	0.010	12

[1]http://mmc36.informatik.uni-augsburg.de/VSSN06 OSAC
[2]http://perception.i2r.a-star.edu.sg/bk model/bk index.htm
[3]http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

D. Performance measures

In order to quantify the performance of each of the alternatives, we employed a total of six measures. First we define A as the set of pixels corresponding to foreground and B as the set of pixels classified as foreground by the segmentation method:

$$A = \{\mathbf{t} \mid \chi(\mathbf{t}) = 1\} \tag{46}$$

$$B = \{\mathbf{t} \mid \widetilde{\chi}(\mathbf{t}) = 1\} \tag{47}$$

Two of the basic measures that we use is the ratio of false negatives (FN) and false positives (FP), which are defined as follow (lower is better):

$$FN = \frac{\operatorname{card}\left(A \cap \overline{B}\right)}{\operatorname{card}\left(A \cup B\right)} \tag{48}$$

$$FP = \frac{\operatorname{card}\left(\overline{A} \cap B\right)}{\operatorname{card}\left(A \cup B\right)} \tag{49}$$

where 'card' stands for the number of elements of a set. Another pair of measures we have used in this study is precision (PR) and recall (RC) (higher is better):

$$PR = \frac{\operatorname{card}\left(A \cap B\right)}{\operatorname{card}\left(B\right)} \tag{50}$$

$$RC = \frac{\operatorname{card}\left(A \cap B\right)}{\operatorname{card}\left(A\right)} \tag{51}$$

$$PR,\ RC \in [0, 1] \tag{52}$$

Sometimes is hard to compare pairs of PR and RC values in order to know what determine better results, so we can use F-measure (FM) because it combines both in one measure (higher is better):

$$FM = \frac{2 \cdot PR \cdot RC}{PR + RC} \tag{53}$$

Another interesting measure is accuracy (AC) defined as follows (higher is better):

$$AC = \frac{\operatorname{card}\left(A \cap B\right)}{\operatorname{card}\left(A \cup B\right)} \tag{54}$$

E. Qualitative results

In this section we will discuss qualitatively some results. We will use Figure 1 in which we show some frames using the best configuration for each color space according to F-measure.

It is generally observed that both RGB and YCbCr are the best performing color spaces. The artifact most frequently observed is camouflage, this occurs when the background and foreground colors are so similar that segmentation algorithms take as background pixels that are actually foreground.

On the other hand we can say that Lab and Luv achieve similar results, although Luv gets more false negatives (FN). This effect is especially remarkable in Meeting Room, Subway Station and Video4.

Finally HSV is an option that achieves good results in both Campus and Lobby. However it makes a large number of false positives (FP) in other sequences like Corridor, Video2, Video4 and LightSwitch.

F. Quantitative results

The performance of color spaces has been different in each sequence. To help us know what is best in each case, we can use Figure 2 which shows the performance of the best configuration for each sequence in terms of F-measure.

First we observe that HSV achieves good results in Campus and Lobby sequences, although in Lobby it can be noticed that the performance is different in the first half of the video, before a series of sudden changes in illumination happens. The RGB space overcomes the other color spaces in 4 sequences: Fountain, levelCrossing, Corridor and LightSwitch. It also obtains very competitive results in Video2 and WaterSurface. However YCbCr is the best choice in most cases, and it remains as an interesting option in the rest. Good examples of this situation are: Meeting Room, Subway Station, Video4 or WaterSurface.

In order to get a more accurate picture of the performance of each color spaces we use Table II which show the mean accuracy achieved by the best configuration regarding this measure. This table confirms our previous statement that RGB and YCbCr are the best choices.

Here we should note that in most cases (except RGB and HSV) the best results are obtained setting $\alpha = 1$ or close to this value, which indicates that the most important channel for the correct segmentation is which corresponds to the lightness, while the remaining channels often add nothing to the result.

Table II
BEST RESULTS ACCORDING TO ACCURACY. THE FIRST AND SECOND COLUMNS DENOTE THE SEQUENCE NAME AND COLOR SPACE, RESULTS ARE SHOWN IN THE THIRD COLUMN, WHEREAS THE LAST THREE COLUMNS CORRESPOND TO PARAMETERS THAT ACHIEVE THIS PERFORMANCE. THE BEST RESULT FOR EACH SEQUENCE IS SHOWN IN **BOLD**.

Sequence	Color Space	Result	α	β	γ
Campus	RGB	0.42 ± 0.19	0.00	0.00	1.00
	Lab	0.26 ± 0.18	1.00	0.00	0.00
	Luv	0.26 ± 0.19	1.00	0.00	0.00
	HSV	**0.61 ± 0.20**	0.80	0.00	0.20
	YCbCr	0.43 ± 0.10	1.00	0.00	0.00
Meeting Room	RGB	0.29 ± 0.26	1.00	0.00	0.00
	Lab	0.31 ± 0.13	1.00	0.00	0.00
	Luv	0.23 ± 0.16	1.00	0.00	0.00
	HSV	0.29 ± 0.06	1.00	0.00	0.00
	YCbCr	**0.70 ± 0.10**	1.00	0.00	0.00
Subway Station	RGB	0.15 ± 0.14	0.00	1.00	0.00
	Lab	0.38 ± 0.15	1.00	0.00	0.00
	Luv	0.01 ± 0.02	1.00	0.00	0.00
	HSV	0.19 ± 0.10	1.00	0.00	0.00
	YCbCr	**0.47 ± 0.16**	1.00	0.00	0.00
Fountain	RGB	**0.56 ± 0.14**	0.00	0.00	1.00
	Lab	0.23 ± 0.10	1.00	0.00	0.00
	Luv	0.25 ± 0.10	1.00	0.00	0.00
	HSV	0.27 ± 0.11	0.80	0.00	0.20
	YCbCr	0.54 ± 0.14	1.00	0.00	0.00
LevelCrossing	RGB	**0.72 ± 0.06**	0.00	0.00	1.00
	Lab	0.30 ± 0.06	1.00	0.00	0.00
	Luv	0.32 ± 0.07	1.00	0.00	0.00
	HSV	0.50 ± 0.18	0.40	0.60	0.00
	YCbCr	0.69 ± 0.05	1.00	0.00	0.00
Corridor	RGB	**0.73 ± 0.03**	0.00	0.00	1.00
	Lab	0.61 ± 0.02	1.00	0.00	0.00
	Luv	0.60 ± 0.02	1.00	0.00	0.00
	HSV	0.27 ± 0.03	0.33	0.33	0.33
	YCbCr	0.68 ± 0.03	0.80	0.20	0.00
Video2	RGB	0.66 ± 0.13	0.00	1.00	0.00
	Lab	0.35 ± 0.21	1.00	0.00	0.00
	Luv	0.36 ± 0.19	1.00	0.00	0.00
	HSV	0.54 ± 0.10	0.40	0.60	0.00
	YCbCr	**0.69 ± 0.16**	1.00	0.00	0.00
Video4	RGB	0.51 ± 0.07	1.00	0.00	0.00
	Lab	0.45 ± 0.05	1.00	0.00	0.00
	Luv	0.00 ± 0.00	1.00	0.00	0.00
	HSV	0.47 ± 0.09	0.60	0.00	0.40
	YCbCr	**0.63 ± 0.07**	1.00	0.00	0.00
WaterSurface	RGB	0.77 ± 0.04	0.00	1.00	0.00
	Lab	0.71 ± 0.07	1.00	0.00	0.00
	Luv	0.72 ± 0.06	1.00	0.00	0.00
	HSV	0.37 ± 0.05	1.00	0.00	0.00
	YCbCr	**0.78 ± 0.05**	1.00	0.00	0.00
Lobby	RGB	0.15 ± 0.21	1.00	0.00	0.00
	Lab	0.05 ± 0.10	1.00	0.00	0.00
	Luv	0.03 ± 0.10	0.60	0.20	0.20
	HSV	**0.26 ± 0.16**	1.00	0.00	0.00
	YCbCr	0.14 ± 0.09	1.00	0.00	0.00
LightSwitch	RGB	**0.41 ± 0.15**	0.00	1.00	0.00
	Lab	0.15 ± 0.09	1.00	0.00	0.00
	Luv	0.11 ± 0.06	1.00	0.00	0.00
	HSV	0.09 ± 0.09	0.33	0.33	0.33
	YCbCr	0.17 ± 0.14	1.00	0.00	0.00

V. CONCLUSIONS

A common framework to choose the optimal color space and color component scaling for foreground object detection has been presented. It has been applied to a probabilistic background model which is based on a self-organizing neural network. The performance of the proposal has been tested with several well known benchmark videos, and it has been found that the luminance color components are the most relevant ones for this task. This work opens the way to tune other background models with the proposed framework to achieve better object detection performance.

ACKNOWLEDGMENTS

This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grant TIN2011-24141. It is also partially supported by the Autonomous Government of Andalusia (Spain) and the European Regional Development Fund under projects TIC-6213 and TIC-657.

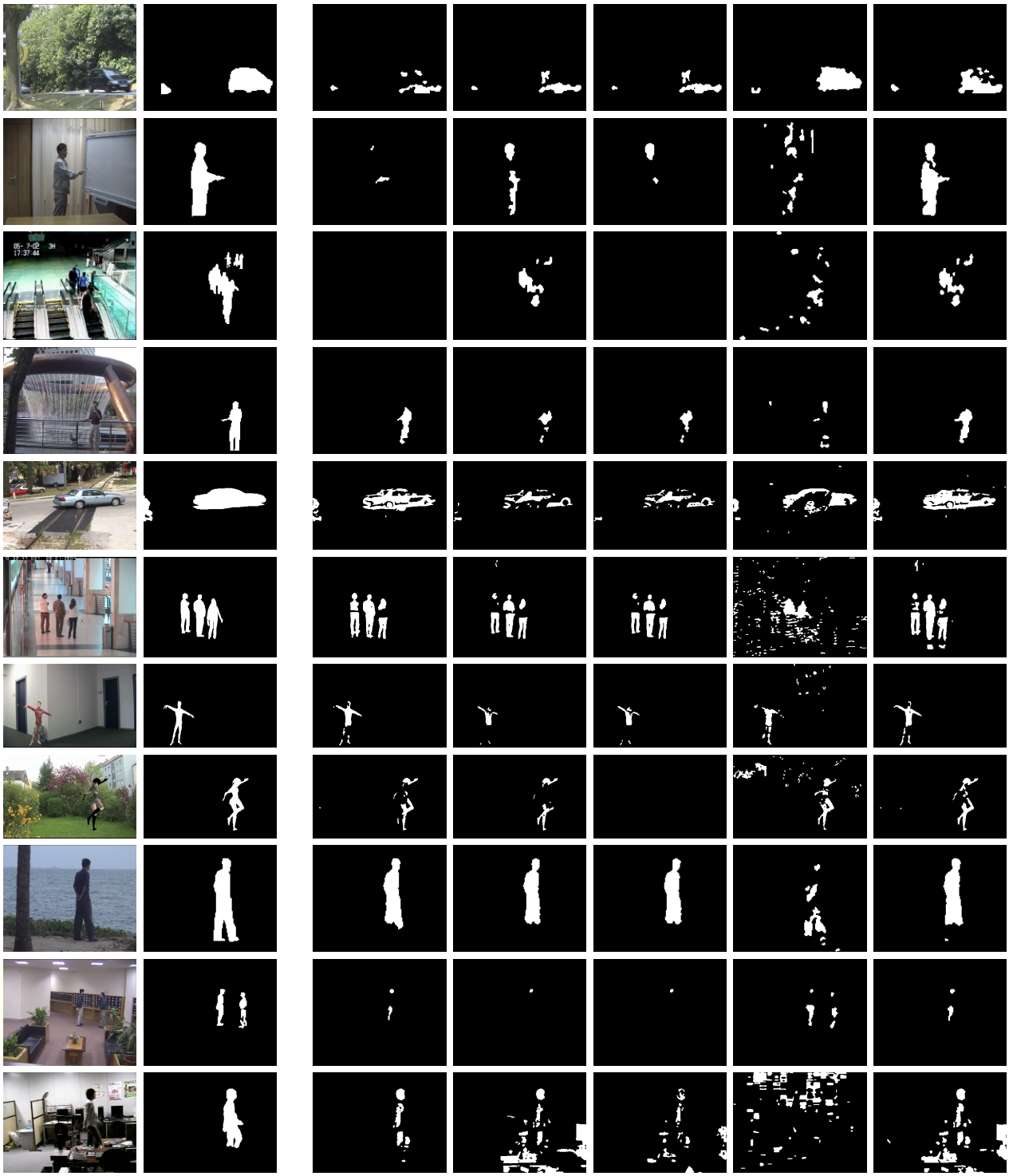

Figure 1. Experimental results. Rows from top to bottom: Campus (frame 2348), Meeting Room (23835), Subway Station (4787), Fountain (1489), LevelCrossing (440), Corridor (370), Video2 (550), Video4 (690), WaterSurface (1559), Lobby (2440) and LightSwitch (1880). The first two columns show original frame and GroundTruth, the last five columns show the tested color spaces: RGB, Lab, Luv, HSV and YCbCr.

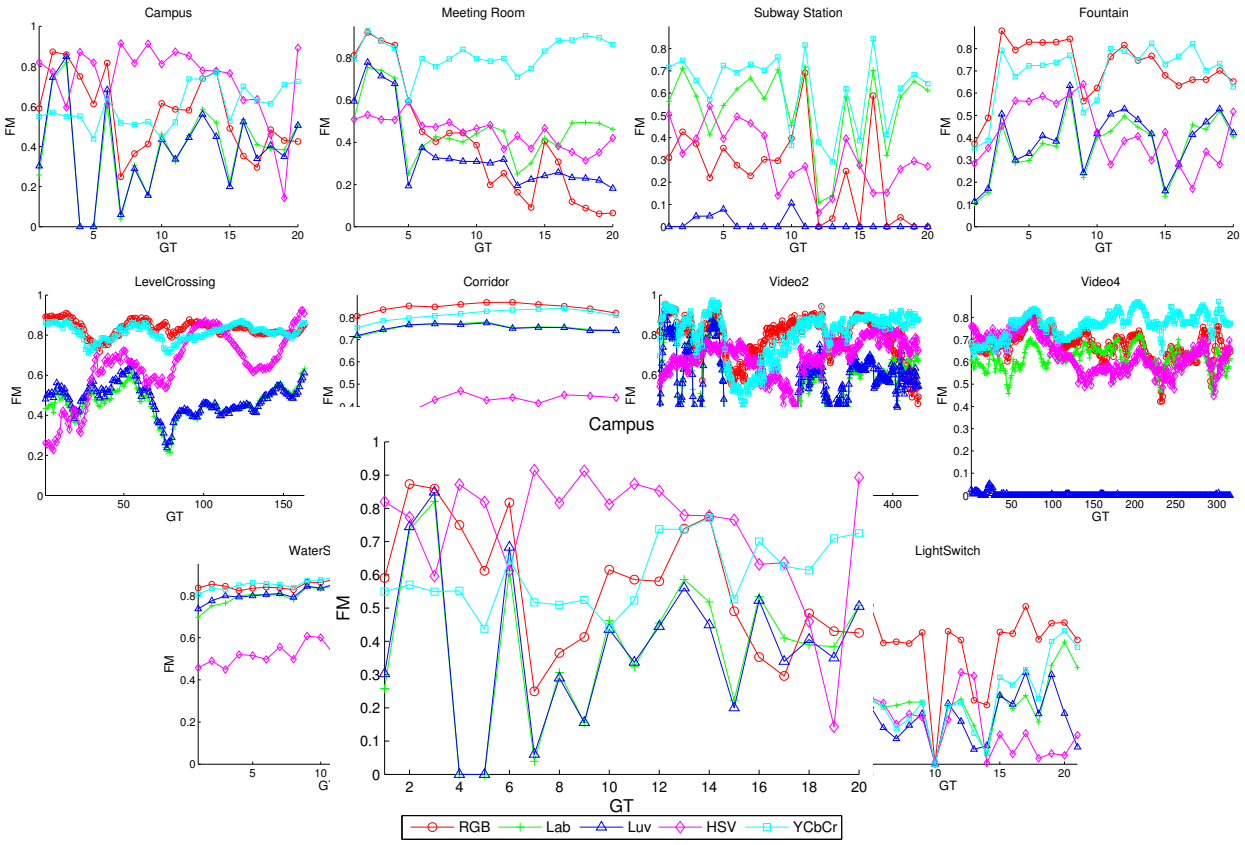

Figure 2. Comparisons between different color spaces using F-measure. The horizontal axis shows the ground truth frame and the vertical axis the F-measure results.

All of them include FEDER funds. 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] Y. Ming and J. Jiang, "Background modeling and moving-objects detection based on Cauchy distribution for video sequence," *Acta Optica Sinica*, vol. 28, no. 3, pp. 587–592, 2008.

[2] J. Ning, Y. Yang, and F. Zhu, "Background modeling and fuzzy clustering for motion detection from video," *Journal of Multimedia*, vol. 8, no. 5, pp. 626–631, 2013.

[3] M. Jlassi, A. Douik, and H. Messaoud, "Color images segmentation algorithms during a sports meeting: Application to soccer video images," *Journal of Circuits, Systems and Computers*, vol. 19, no. 6, pp. 1307–1332, 2010.

[4] T. Gao, Z.-G. Liu, S.-H. Yue, J.-Q. Mei, and J. Zhang, "Traffic video-based moving vehicle detection and tracking in the complex environment," *Cybernetics and Systems*, vol. 40, no. 7, pp. 569–588, 2009.

[5] A. Doshi and M. Trivedi, "Satellite imagery based adaptive background models and shadow suppression," *Signal, Image and Video Processing*, vol. 1, no. 2, pp. 119–132, 2007.

[6] M. Balcilar, F. Karabiber, and A. Sonmez, "Performance analysis of Lab2000HL color space for background subtraction," in *2013 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2013.

[7] E. López-Rubio, R. M. Luque-Baena, and E. Domínguez, "Foreground detection in video sequences with probabilistic self-organizing maps," *International Journal of Neural Systems*, vol. 21, no. 3, pp. 225–246, 2011.

[8] D. H. Brainard, *The Science of Color*. Elsevier, 2003, ch. Color Appearance and Color Difference Specification, pp. 191–216.

[9] C. M. Bishop and M. Svenson, "The generative topographic mapping," *Neural Computation*, vol. 10, no. 1, pp. 215–234, 1998.

[10] M. M. Van Hulle, "Kernel-based topographic map formation by local density modeling," *Neural Computation*, vol. 14, no. 7, pp. 1561–1573, 2002.

[11] M. Van Hulle, "Maximum likelihood topographic map formation," *Neural Computation*, vol. 17, no. 3, pp. 503–513, 2005.

[12] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.

[13] H. J. Kushner and G. G. Yin, *Stochastic approximation and Recursive Algorithms and Applications*. New York, NY, USA: Springer-Verlag, 2003.

[14] E. López-Rubio, J. M. Ortiz-de-Lazcano-Lobato, and D. López-Rodríguez, "Probabilistic PCA self-organizing maps," *IEEE Trans. on Neural Networks*, vol. 20, no. 9, pp. 1474–1489, 2009.

[15] E. López-Rubio, "Multivariate Student-t self-organizing maps," *Neural Networks*, vol. 22, no. 10, pp. 1432–1447, 2009.

[16] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1168–1177, 2008.

[17] E. López-Rubio, "Robust location and spread measures for nonparametric probability density function estimation," *International Journal of Neural Systems*, vol. 19, no. 5, pp. 345–357, 2009.