A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications

Lucia Maddalena and Alfredo Petrosino, *Senior Member, IEEE*

***Abstract*—Detection of moving objects in video streams is the first relevant step of information extraction in many computer vision applications. Aside from the intrinsic usefulness of being able to segment video streams into moving and background components, detecting moving objects provides a focus of attention for recognition, classification, and activity analysis, making these later steps more efficient. We propose an approach based on self organization through artificial neural networks, widely applied in human image processing systems and more generally in cognitive science. The proposed approach can handle scenes containing moving backgrounds, gradual illumination variations and camouflage, has no bootstrapping limitations, can include into the background model shadows cast by moving objects, and achieves robust detection for different types of videos taken with stationary cameras. We compare our method with other modeling techniques and report experimental results, both in terms of detection accuracy and in terms of processing speed, for color video sequences that represent typical situations critical for video surveillance systems.**

***Index Terms*—Background subtraction, motion detection, neural network, self organization, visual surveillance.**

I. Introduction

VISUAL surveillance is a very active research area in computer vision thanks to the rapidly increasing number of surveillance cameras that leads to a strong demand for automatic processing methods for their output. The scientific challenge is to devise and implement automatic systems able to detect and track moving objects, and interpret their activities and behaviors. The need is strongly felt world-wide, not only by private companies, but also by governments and public institutions, with the aim of increasing people safety and services efficiency. Visual surveillance is indeed a key technology for fight against terrorism and crime, public safety (e.g., in transport networks, town centers, schools, and hospitals), and efficient management of transport networks and public facilities (e.g., traffic lights, railroad crossings) [1].

The main tasks in visual surveillance systems include motion detection, object classification, tracking, activity understanding, and semantic description. Our focus here is on the detection phase of a general visual surveillance system using static cameras. The detection of moving objects in video streams is the first relevant step of information extraction in many computer vision applications. Aside from the intrinsic usefulness of being able to segment video streams into foreground and background components, detecting moving objects provides a focus of attention for recognition, classification, and activity analysis, making these later steps more efficient, since only moving pixels need be considered [2].

The usual approach to moving object detection is through background subtraction, that consists in maintaining an up-to-date model of the background and detecting moving objects as those that deviate from such a model. Compared to other approaches, such as optical flow (e.g., [3]), this approach is computationally affordable for real-time applications. The main problem is its sensitivity to dynamic scene changes, and the consequent need for the background model adaptation via background maintenance. Such problem is known to be significant and difficult [4]. Some of the well-known issues in background maintenance, that will be specifically addressed in the sequel, include:

- *light changes*: the background model should adapt to gradual illumination changes;
- *moving background*: the background model should include changing background that is not of interest for visual surveillance, such as waving trees;
- *cast shadows*: the background model should include the shadow cast by moving objects that apparently behaves itself moving, in order to have a more accurate detection of the moving objects shape;
- *bootstrapping*: the background model should be properly set up even in the absence of a complete and static (free of moving objects) training set at the beginning of the sequence;
- *camouflage*: moving objects should be detected even if their chromatic features are similar to those of the background model.

Our approach to moving object detection is based on the background model automatically generated by a self-organizing method without prior knowledge about the involved patterns. The idea consists in adopting biologically inspired methods for moving object detection, where visual attention mechanisms are used to help detecting objects that keep the user attention in accordance with a set of predefined features, such as gray level, motion, and shape features. It will be shown, by qualitative and quantitative results, that our adaptive model, extending [5], can cope with all the above mentioned issues for background

Manuscript received November 13, 2007; revised March 26, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Til Aach.

L. Maddalena is with the Institute for High-Performance Computing and Networking, National Research Council, 80131 Naples, Italy (e-mail: lucia.maddalena@na.icar.cnr.it).

A. Petrosino is with the Department of Applied Science, University of Naples Parthenope, 80143 Naples, Italy (e-mail: alfredo.petrosino@uniparthenope.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2008.924285

maintenance and achieves robust detection for different types of videos taken with stationary cameras.

The paper is organized as follows. In Section II, we present a fairly compact overview of existing approaches adopted for background subtraction. Section III reports the motivations and the background methodologies our algorithm relies on. In Section IV, we present results achieved with the implementation of the proposed approach in terms of attained accuracy and efficiency, comparing them with those obtained by several other existing methods. Section V includes conclusions and further research directions.

II. Related Work

Due to its pervasiveness in various contexts, background subtraction has been afforded by several researchers, and plenty of literature has been published (see surveys in [6]–[8]). There is no unique classification of proposed methods. Some usually referred dichotomies, here cited in order to highlight advantages and tradeoffs of most existing methods, include the following.

- **Parametric versus nonparametric**: Parametric models (e.g., [9]) are tightly coupled with underlying assumptions, not always perfectly corresponding to the real data, and the choice of parameters can be cumbersome, thus reducing automation. On the other hand, nonparametric models (e.g., [10] and [11]) are more flexible but heavily data dependent.
- **Unimodal versus multimodal**: Basic background models assume that the intensity values of a pixel can be modeled by a single unimodal distribution (e.g., [9]). Such models usually have low complexity, but cannot handle moving backgrounds, while this is possible with multimodal models (e.g., [11] and [12]), at the price of higher complexity.
- **Recursive versus nonrecursive**: Nonrecursive techniques (e.g., [4], [13], [14]) store a buffer of a certain number of previous sequence frames and estimate the background model based on the temporal variation of each pixel within the buffer, while recursive techniques (e.g., [9] and [12]) recursively update a single background model based on each input frame. In the first case, the background well adapts to eventual variations, but memory requirements can be significant; in the latter, space complexity is lower, but input frames from distant past can have an effect on the current background, and, therefore, any error in the background model is carried out for a long time period.
- **Pixel-based versus region-based**: Pixel-based methods (e.g., [9], [11], [12], [14]) assume that the time series of observations is independent at each pixel, while region-based methods (e.g., [2], [4], [10]) take advantage of interpixel relations, segmenting the images into regions or refining the low-level classification obtained at the pixel level. This step obviously increases the overall complexity.

Our approach is based on the background model automatically generated by a self-organizing method and can be broadly classified as nonparametric, multimodal, recursive, and pixel-based.

III. Modeling the Background by Self-Organization

As already pointed out in Section I, the main problem of the background subtraction approach to moving object detection is its extreme sensitivity to dynamic scene changes due to lighting and extraneous events. Although these are usually detected, they leave behind "holes" where the newly exposed background imagery differs from the known background model (ghosts). While the background model eventually adapts to these "holes," they generate false alarms for a short period of time. Therefore, it is highly desirable to construct an approach to motion detection based on a background model that automatically adapts to changes in a self-organizing manner and without *a priori* knowledge.

We propose to adopt a biologically inspired problem-solving method based on visual attention mechanisms. The aim is to obtain the objects that keep the user attention in accordance with a set of predefined features, including gray level, motion and shape features (see for instance [15]–[17]). Our approach defines a method for the generation of an active attention focus to monitor dynamic scenes for surveillance purposes. The idea is to build the background model by learning in a self-organizing manner many background variations, i.e., background motion cycles, seen as trajectories of pixels in time. Based on the learnt background model through a map of motion and stationary patterns, our algorithm can detect motion and selectively update the background model. Specifically, a novel neural network mapping method is proposed to use a whole trajectory incrementally in time fed as an input to the network. This makes the network structure much simpler and the learning process much more efficient.

The adopted artificial neural network is organized as a 2-D flat grid of neurons (or nodes) and, similarly to self-organizing maps (SOMs) or Kohonen networks [18], allows to produce representations of training samples with lower dimensionality, at the same time preserving topological neighborhood relations of the input patterns (nearby outputs correspond to nearby input patterns). Each node computes a function of the weighted linear combination of incoming inputs, where weights resemble the neural network learning. Doing so, each node could be represented by a weight vector, obtained collecting the weights related to incoming links. In the following, the set of weight vectors will be called a *model*. An incoming pattern is mapped to the node whose model is "most similar" (according to a predefined metric) to the pattern, and weight vectors in a neighborhood of such node are updated. Therefore, the network behaves as a competitive neural network that implements a winner-take-all function with an associated mechanism, that modifies the local synaptic plasticity of the neurons, allowing learning to be restricted spatially to the local neighborhood of the most active neurons. For each color pixel, we consider a neuronal map consisting of $n \times n$ weight vectors. Each incoming sample is mapped to the weight vector that is closest according to a suitable distance measure, and the weight vectors in its neighborhood are updated. The whole set of weight vectors acts as a background model, that is used for background subtraction in order to identify moving pixels.

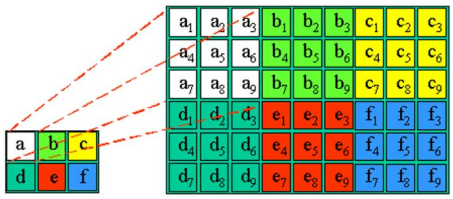

Fig. 1. (Left) Simple image and the (right) neuronal map structure.

A. Initial Background Model

In the case of our background modeling application, we have at our disposal a fairly good means of initializing the weight vectors of the network: the first image of our sequence is indeed a good initial approximation of the background, and, therefore, for each pixel, the corresponding weight vectors are initialized with the pixel value.

In order to represent each weight vector, we choose the HSV color space, relying on the hue, saturation and value properties of each color. Such color space allows us to specify colors in a way that is close to human experience of colors. Moreover, the intensity of the light is explicit and separated from chromaticity, and this allows change detection invariant to modifications of illumination strength. Let (h, s, v) be the HSV components of the generic pixel (x, y) of the first sequence frame I_0, and let $C = (c_1, c_2, \ldots, c_{n^2})$ be the model for pixel (x, y). Each weight vector $c_i, i = 1, \ldots, n^2$, is a 3-D vector initialized as $c_i = (h, s, v)$.

The complete set of weight vectors for all pixels of an image I with N rows and M columns is represented as a neuronal map A with $n \times N$ rows and $n \times M$ columns, where the weight vectors for the generic pixel (x, y) of I are at neuronal map positions $(i, j), i = n \times x, \ldots, n \times (x + 1) - 1$ and $j = n \times y, \ldots, n \times (y + 1) - 1$. An example of such neuronal map structure for a simple image I with $N = 2$ rows and $M = 3$ columns obtained choosing $n = 3$ is given in Fig. 1. As depicted, the upper left pixel a of I (Fig. 1, left) has weight vectors $(a_1, \ldots, a_9)$ stored into the 3×3 elements of the upper left part of neuronal map A (Fig. 1, right). Analogous relations exist for each pixel of I and corresponding weight vectors storage. It appears evident that the neuronal map A can be seen as an initial background model, enlarged 3×3 times.

This configuration allows to easily take into account the spatial relationship among pixels and corresponding weight vectors, as we shall see in the following section.

B. Subtraction and Update of the Background Model

After initialization, temporally subsequent samples are fed to the network. Each incoming pixel p_t of the tth sequence frame I_t is compared to the current pixel model C to determine if there exists a weight vector that best matches it. If a best matching weight vector c_m is found, it means that p_t belongs to the background and it is used as the pixel encoding approximation, and the best matching weight vector, together with its neighborhood, is reinforced. Otherwise, if no acceptable matching weight vector exists, we discriminate whether p_t is in the shadow cast by some object or not. In the first case, p_t should be still considered as background, but it should not be used to update the corresponding weight vectors, in order to avoid the reinforcement of shadow information into the background model; in the latter case p_t is detected as belonging to a moving object (foreground).

The above described background subtraction and update procedure for each pixel can be sketched as in the following algorithm.

Algorithm SOBS (Self-Organizing Background Subtraction)

```
Input: pixel value p_t in frame I_t, t = 0,...,LastFrame
Output: background/foreground binary mask value B(p_t)
1. Initialize model C for pixel p_0 and store it into A
2. for t = 1, LastFrame
3.  Find best match c_m in C to current sample p_t
4.  if (c_m found) then
5.   B(p_t) = 0//background
6.   update A in the neighborhood of c_m
7.  else if(p_t shadow) then
8.   B(p_t) = 0//background
9.  else
10.  B(p_t) = 1//foreground
```

In practice, we distinguish the whole process into two phases: a *calibration* phase and an *online* phase. The *calibration* phase involves the neural network initial learning, and consists in steps 1–6 executed on the first $K + 1$ sequence frames (i.e., for $t = 1, K$ in step 2), with $K < \mathrm{LastFrame}$. The *online* phase involves neural network adaptation and background subtraction, and consists in steps 2–10 executed on the last $\mathrm{LastFrame} - K$ sequence frames (i.e., for $t = K + 1, \mathrm{LastFrame}$ in step 2). The number K of sequence frames for the calibration phase basically depends on how many *static* (free of moving foreground objets) initial frames are available for each sequence.

Steps 3, 6, and 7 of SOBS algorithm will next be described in detail.

1) Finding the Best Match C_m in C To Current Sample P_t: To determine which weight vector gives the best match, several metrics for detecting changes in color imagery, such as those reported in [19]–[21] and in references therein, could be adopted. Experiments lead us to employ the Euclidean distance of vectors in the HSV color hexcone [19], that gives the distance between two pixels $p_i = (h_i, s_i, v_i)$ and $p_j = (h_j, s_j, v_j)$ as

$$d(p_i, p_j) = \|(v_i s_i \cos(h_i), v_i s_i \sin(h_i), v_i) - (v_j s_j \cos(h_j), v_j s_j \sin(h_j), v_j)\|_2^2. \quad (1)$$

Indeed, the representation of HSV values as vectors in the HSV color hexcone used in such distance measure allows to avoid problems connected with the periodicity of hue (that represents an angle) and with the instability of hue for small values of saturation (hue is undefined for null saturation) [19].

Weight vector c_m, for some m, gives the best match for the incoming pixel p_t if its distance from p_t is minimum in the model C of p_t and is no greater than a fixed threshold

$$d(c_m, p_t) = \min_{i=1,\ldots,n^2} d(c_i, p_t) \leq \epsilon.$$

The threshold ϵ allows to distinguish between foreground and background pixels, and is chosen as

$$\epsilon = \begin{cases} \epsilon_1, & \text{if } 0 \leq t \leq K \\ \epsilon_2, & \text{if } t > K \end{cases} \quad (2)$$

with ϵ_1 and ϵ_2 small constants. Specifically, ϵ_1 should be greater than ϵ_2, since higher values for ϵ_1 allow, within the first K sequence frames, to obtain a (possibly rough) background model including several observed pixel intensity variations, while lower values for ϵ_2 allow to obtain a more accurate background model in the online phase.

2) Updating A in the Neighborhood of c_m: If a best match c_m is found for current sample p_t, the weight vectors in the $n \times n$ neighborhood of c_m are updated according to selective weighted running average. In details, given the incoming pixel $p_t(x, y)$ at spatial position (x, y) and time t, if there exists a best match c_m in the model C of p_t, and c_m is present in the background model at position $(\overline{x}, \overline{y})$, then weight vectors A_t in the $n \times n$ neighborhood of $(\overline{x}, \overline{y})$ are updated according to

$$A_t(i, j) = (1 - \alpha_{i,j}(t))A_{t-1}(i, j) + \alpha_{i,j}(t)p_t(x, y) \quad (3)$$

for $i = \overline{x} - \lfloor n/2 \rfloor, \ldots, \overline{x} + \lfloor n/2 \rfloor$, and $j = \overline{y} - \lfloor n/2 \rfloor, \ldots, \overline{y} + \lfloor n/2 \rfloor$. For the example neuronal map reported in Fig. 1, if the best match for current pixel a is the weight vector a_6, then the weight vectors that are updated according to (3) are weight vectors $a_2, a_3, b_1, a_5, a_6, b_4, a_8, a_9, b_7$, that belong in part to the model of pixel a and in part to the model of pixel b of current image I. Therefore, such updating allows to take into account spatial relationships among incoming pixel with its surrounding.

In (3), $\alpha_{i,j}(t) = \alpha(t)w_{i,j}$, where $w_{i,j}$ are Gaussian weights in the $n \times n$ neighborhood, that well correspond to the lateral inhibition activity of neurons. Moreover, $\alpha(t)$ represents the learning factor, chosen as

$$\alpha(t) = \begin{cases} \alpha_1 - t\frac{\alpha_1 - \alpha_2}{K}, & \text{if } 0 \leq t \leq K \\ \alpha_2, & \text{if } t > K \end{cases}$$

where α_1 and α_2 are predefined constants such that $\alpha_2 \leq \alpha_1$. This means that during the calibration phase we choose the learning factor $\alpha(t)$ as a monotonically decreasing function of time t, in order to ensure neural network convergence as stated in [18], while during the subsequent online phase we choose it as a constant value that depends on the scene variability. Specifically, large α_2 values enable the network to faster learn changes corresponding to background, but leading to faster inclusion into the background model of pixels belonging to foreground moving objects that have erroneously been detected as background (false negatives). On the other hand, lower learning rates make the network slower to adapt to rapid background changes, but making the model more tolerant to errors due to false negatives. Indeed in this case the problem is more easily corrected through self-organization, since weight vectors of false negative pixels are readily smoothed out by the learning process itself. In order to have in (3) values for $\alpha_{i,j}(t)$ that belong to $[0, 1]$, α_1 and α_2 are chosen as

$$\alpha_1 = \frac{c_1}{\max\{w_{i,j}\}}$$
$$\alpha_2 = \frac{c_2}{\max\{w_{i,j}\}} \quad (4)$$

with c_1 and c_2 predefined constants such that $0 \leq c_2 \leq c_1 \leq 1$.

If the best match c_m is not found, the background model A remains unchanged. This selectivity allows to adapt the background model to scene modifications (including gradual light changes) without introducing the contribution of pixels that do not belong to the background scene.

3) Shadow Detection: The approach adopted for moving cast shadow detection is adapted from that reported in [13], that proved to be quite accurate and suitable for moving object detection. The basic idea is that a cast shadow darkens the background, while a moving object can darken it or not depending on its color. Specifically, in a shadowed area there is a significant illumination variation, but only a small color variation. A pixel p_t of I_t belongs to the shadow cast by an object in the scene if it belongs to the background model but has been darkened by a shadow, i.e., if there exists at least one weight vector c_i belonging to the model $C = (c_1, \ldots, c_{n^2})$ of p_t such that

$$\left(\gamma \leq \frac{p_t^V}{c_i^V} \leq \beta\right) \wedge \left(p_t^S - c_i^S \leq \tau_S\right) \wedge \left(|p_t^H - c_i^H| \leq \tau_H\right) \quad (5)$$

where p_t^H, p_t^S, and p_t^V indicate the hue, saturation, and value components of pixel p_t, and analogous notation has been adopted for HSV components of the weight vector c_i. A thorough discussion about parameters γ, β, τ_S, and τ_H, together with suitable values for them, is reported in [13] and references therein.

IV. Experimental Results

A. Data and Qualitative Results

Experimental results for moving object detection using the proposed approach have been produced for several image sequences. Here, we describe five different sequences, that represent typical situations critical for video surveillance systems, and present qualitative results obtained with the proposed method.

A detailed discussion about SOBS algorithm parameters selection should be made.

- The number n^2 of weight vectors used to model each pixel has been fixed to 9 for all the reported experiments. This choice has been driven by experiments carried out, where, varying n, we observed almost constant accuracy, but rapidly growing execution times.
- Values for the distance thresholds ϵ_1 and ϵ_2 in (2), as pointed out in Section III-B1, should be chosen such that $\epsilon_1 \geq \epsilon_2$. High values for ϵ_1 allow to limit selectivity in the update of the background model during the calibration phase, enabling the inclusion into the initial background model of several observed pixel intensity variations.

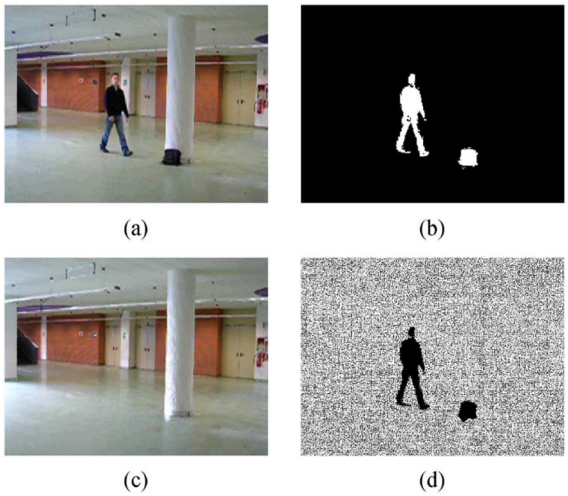

Fig. 2. Results of SOBS algorithm on sequence MSA: (a) original frame; (b) computed moving object detection mask; (c) background model; (d) background model change mask from previous frame.

Lower values for ϵ_2 should be chosen to obtain a more accurate background model in the online phase. Typical choices are $\epsilon_1 \in [0.05, 0.2]$ and $\epsilon_2 \simeq 10^{-1}\epsilon_1$; they will be detailed for each single sequence.

- The number K of sequence frames for the calibration phase depends on how many static initial frames are available for each sequence. If no static initial frames are available, then a sufficiently high value for K should be chosen, together with a high value for ϵ_1 (see previous item), in order to filter out foreground objects from the initial background model, through the weighted running average (3) with limited selectivity. Values for K will be detailed for each single sequence.
- Learning factor c_1 in (4) has been fixed to 1 for all the reported experiments, while c_2, that depends on scene variability, has been experimentally chosen and will be detailed for each single sequence.
- Parameter values in (5) for shadow detection have been experimentally chosen as $\gamma = 0.7, \beta = 1.0, \tau_S = 0.1, \tau_H = 10$ for all the reported experiments.

1) Sequence MSA: Sequence MSA is a home-made indoor sequence manually labeled, consisting of 528 frames of 320×240 spatial resolution, acquired at a frequency of 30 fps (frames per second), made publicly available in the download section of http://cvprlab.uniparthenope.it. The scene consists of a university hall, where a man comes in, leaves a bag on the floor, and then comes out. It represents an example of *easy* sequence, in that lighting conditions are quite stable and moving objects are well contrasted with the background (there is no camouflage); however, strong shadows cast by moving objects can be observed in the entire sequence.

In Fig. 2, we report one of the sequence frames [Fig. 2(a)] and the corresponding moving object detection mask computed by the SOBS algorithm with $K = 45, \epsilon_1 = 0.1, \epsilon_2 = 0.02, c_2 = 0.01$ [Fig. 2(b)]. The detection mask shows that the walking man is perfectly detected, and that most of the shadows cast by the man have been incorporated into the background. It should be observed that the bag, that has been left by the man in previous frames, is still detected as an object extraneous to the background, since SOBS algorithm selective update prevents its inclusion into the background model.

Fig. 3. Results of SOBS algorithm on sequence Walk1: (a) original frame; (b) computed moving object detection mask.

In Fig. 2, we also show the background model A computed by the SOBS algorithm [Fig. 2(c)] and its change mask from previous frame [Fig. 2(d)], where white pixels indicate all weight vectors that have been updated from the previous iteration. We would remark that the background model A is represented by a neuronal map whose size is nine $(n = 3)$ times greater than that of the original image I (in the reported figures they appear to have the same size only for space constraints and for an easier comparison). We can observe that the background model is a quite accurate representation of the real background.

2) Sequence Walk1: Sequence Walk1 of the CAVIAR Project [22] is labeled and comprise 611 frames of 384×288 spatial resolution, captured at a frequency of 25 fps. The scene consists in a laboratory where a man comes in, walks around, and leaves, while small groups of people stand on the left and the bottom side of the lab. This is an example of *hard* sequence, in that lighting conditions are much worse than in the previous, and moving people tend to camouflage with the pavement.

One representative frame is reported in Fig. 3(a), while the corresponding moving object detection mask computed by the SOBS algorithm with $K = 15, \epsilon_1 = 0.1, \epsilon_2 = 0.006, c_2 = 0.05$ is reported in Fig. 3(b). By looking at the detection mask, we can observe that the man in the room center is almost perfectly detected, despite his camouflage with the pavement. This is due to the fact that our model learns background motion trajectories captured by different weight vectors and, therefore, different color clusters, and to the adoption of distance (1), which allows a quite fair discrimination among colors. The group of persons lying close to the lower left side of the image is partially detected. This is reasonable since such persons are barely distinguishable also for the human eye. Same observations hold for the person lying close to the plant in lower center side of the image.

3) Sequence ToD: Sequence ToD (*Time of Day*) belongs to a set of sequences that represent some of the *canonical problems* for background subtraction highlighted in [4]. It has been chosen in order to test our method adaptivity to gradual lighting variations. Here, the scene consists in an empty room, where light gradually brightens, simulating the moving sun; finally, a man comes in and sits on a sofa. The sequence contains 5890 frames of 160×120 spatial resolution, captured at a frequency of 15 fps. Hand-segmented background is given for just one test

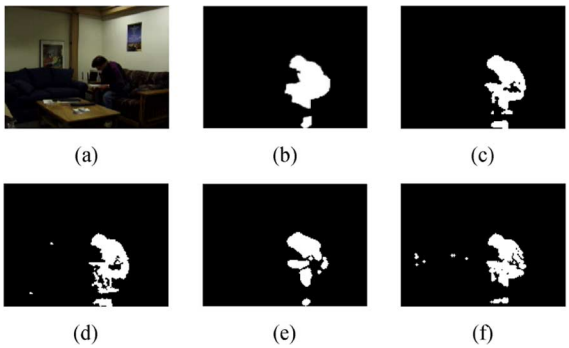

Fig. 4. Segmentation of sequence ToD: (a) test image; (b) ground truth; (c) SOBS result; (d) Pfinder result; (e) VSAM result; (f) CB result.

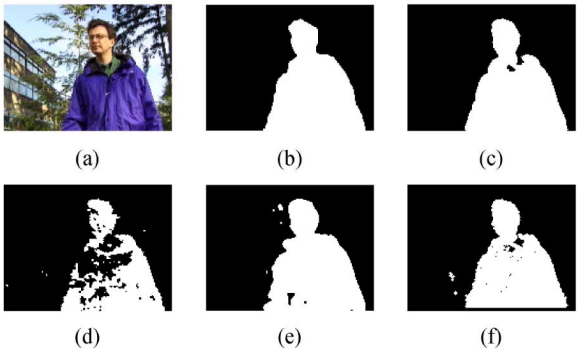

Fig. 5. Segmentation of sequence WT: (a) test image; (b) ground truth; (c) SOBS result; (d) Pfinder result; (e) VSAM result; (f) CB result.

frame, allowing to compare the obtained results on a pixel-by-pixel basis.

The test image and the related ground truth are reported in Fig. 4(a) and (b), respectively. Comparing the ground truth with the foreground mask computed by SOBS algorithm with $K = 200, \epsilon_1 = 0.1, \epsilon_2 = 0.0005, c_2 = 0.1$, reported in Fig. 4(c), we can observe that detection accuracy is quite appreciable, even though the test image is very dark and the man tends to camouflage with the sofa.

4) Sequence WT: Like the previous one, also sequence WT (*Waving Trees*) comes from sequences adopted in [4]. Here, it has been chosen in order to test our method ability to cope with moving background. The outdoor scene includes trees moving in the background and, finally, a man passing in front of the camera; here we are not interested in the waving trees, but only in extraneous moving objects (the man). The sequence contains 287 frames of 160×120 spatial resolution, captured at a frequency of 15 fps. Hand-segmented background [Fig. 5(b)] is given for just one test frame [Fig. 5(a)]. The foreground mask computed by SOBS algorithm with $K = 200, \epsilon_1 = 0.1, \epsilon_2 = 0.03, c_2 = 0.05$ is reported in Fig. 5(c). Here, it can be observed that SOBS algorithm was quite successful in modeling the waving trees as background, and that the walking man has been almost perfectly detected.

5) Sequence IR: Sequence IR (*Intelligent Room*) comes from sequences publicly available at http://cvrr.ucsd.edu/aton/shadow/index.html. The indoor scene consists in an initially empty meeting room, where a man comes in and walks around. Such sequence has been chosen in order to compare accuracy

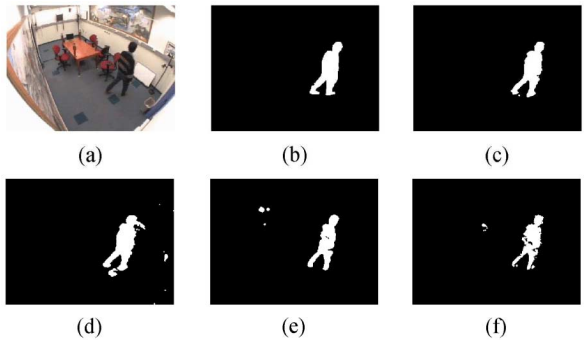

Fig. 6. Segmentation of sequence IR: (a) test image; (b) ground truth; (c) SOBS result; (d) Pfinder result; (e) VSAM result; (f) CB result.

results on a more comprehensive set of ground truths. In fact, it consists of 300 frames of 320×240 spatial resolution, and hand-segmented background masks are available for 113 frames spread along the sequence. It should be mentioned that, while original color ground truths segment the images into background, foreground and shadow regions, the adopted ground truths have been obtained as binary images containing only foreground (white) pixels (corresponding to original foreground) and background (black) pixels (corresponding to original background and shadow regions). One of the test images and the related ground truth are reported in Fig. 6(a) and (b), respectively. The corresponding foreground mask computed by SOBS algorithm with $K = 70, \epsilon_1 = 0.05, \epsilon_2 = 0.004, c_2 = 0.01$ is reported in Fig. 6(c). From such results, it can be observed that SOBS algorithm was quite successful in modeling the background, and that the walking man has been almost perfectly detected.

B. Quantitative Evaluation

Results obtained by the proposed SOBS algorithm have been compared with those obtained by other existing algorithms, in terms of different metrics. Compared methods, adopted metrics and accuracy results will be briefly described in the following.

1) Methods Considered for Comparison: Compared methods will be referred to as *Pfinder, VSAM*, and *CB*.

In the *Pfinder* system, described in [9], the background subtraction algorithm assumes that the pixel intensity values can be modeled by a Gaussian distribution $N(\mu, \sigma^2)$. Running average is used to selectively update μ and σ, while foreground pixels are determined using a threshold Th on the Mahalanobis distance ($|I - \mu|/\sigma > Th$). This is an example of parametric, unimodal, recursive, and pixel-based method, and it has been chosen to represent a well settled basic motion detection procedure.

The *VSAM* system, proposed in [2], is an example of complete visual surveillance system. The moving object detection phase is here based on the integration of pixel analysis and region analysis modules to extract motion by a finite state machine. This method is parametric, unimodal, nonrecursive and both pixel-based and region-based, and it has been chosen here as an advanced reference motion detection system, able to recognize when objects have stopped and disambiguate overlapping objects.

The *CB* algorithm, reported in [11], has been considered here since it presents few similarities with the proposed SOBS approach. Here vector quantization is used to incrementally construct a codebook in order to generate a background model; the best match is found based on a color distortion measure and brightness bounds. According to the dichotomies presented in Section II, the algorithm can be classified as nonparametric, multimodal, recursive and pixel-based.

For all these algorithms, we experimented with different settings of adjustable parameters until the results seemed optimal over the entire sequence.

2) Accuracy Metrics: For measuring accuracy we adopted different metrics, namely *Precision, Recall*, F_1, and *Similarity*.

Recall, also known as *detection rate*, gives the percentage of detected true positives as compared to the total number of true positives in the ground truth

$$\text{Recall} = \frac{tp}{tp + fn}$$

where tp is the total number of *true positives*, fn is the total number of *false negatives*, and $(tp + fn)$ indicates the total number of items present in the ground truth. It should be observed that, if the ground truth is given as a binary detection mask (as, for example, for sequences belonging to [4]), *true positives* (resp. *false negatives*) are to be intended as true positive (resp. false negative) pixels of a single image, and *Recall* acts as a pixel-based measure. If the ground truth is given as information on the number and features of moving objects in the sequence (as, for example, for sequences belonging to [22]), *true positives* (resp. *false negatives*) are to be intended as true positive (resp. false negative) objects in the whole sequence, and *Recall* acts as a frame-based metric.

Recall alone is not enough to compare different methods, and is generally used in conjunction with *Precision*, also known as *positive prediction*, that gives the percentage of detected true positives as compared to the total number of items detected by the method

$$\text{Precision} = \frac{tp}{tp + fp}.$$

Here, fp is the total number of *false positives*, $(tp + fp)$ indicates the total number of detected items, and considerations analogous to those reported for *Recall* should be made to clarify the pixel-based or frame-based approach of the adopted metric.

Using the above mentioned metrics, generally, a method is considered *good* if it reaches high *Recall* values, without sacrificing *Precision*.

Moreover, we considered the F_1 metric, also known as *Figure of Merit* or *F-measure*, that is the weighted harmonic mean of *Precision* and *Recall*

$$F_1 = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}.$$

Such measure allows to obtain a single measure that can be used to "rank" different methods.

Finally, we considered the pixel-based *Similarity* measure, defined as

$$\text{Similarity} = \frac{tp}{tp + fn + fp}.$$

TABLE I
FRAME-BASED ACCURACY VALUES FOR SEQUENCES MSA AND WALK1

	MSA			Walk1		
	Recall	Precision	F_1	Recall	Precision	F_1
SOBS	0.99	0.99	0.99	0.76	0.76	0.76
Pfinder	0.93	0.89	0.91	0.60	0.57	0.58
VSAM	0.99	0.98	0.98	0.68	0.82	0.74
CB	0.98	0.99	0.98	0.88	0.60	0.71

TABLE II
PIXEL-BASED ACCURACY VALUES FOR SEQUENCE TOD

	Recall	Precision	F_1	Similarity
SOBS	0.73	0.97	0.83	0.71
Pfinder	0.69	0.86	0.77	0.62
VSAM	0.61	0.95	0.75	0.60
CB	0.72	0.95	0.82	0.69

TABLE III
PIXEL-BASED ACCURACY VALUES FOR SEQUENCE WT

	Recall	Precision	F_1	Similarity
SOBS	0.97	0.98	0.98	0.96
Pfinder	0.67	0.97	0.79	0.66
VSAM	0.99	0.94	0.96	0.93
CB	0.97	0.97	0.97	0.94

This metric, that is analogous to the F_1 metric, has been adopted with the only objective of further comparing the results achieved by the proposed SOBS algorithm with those reported in [23] (see Section IV-B4).

All the above considered measures attain values in [0, 1], and the higher is the value, the better is the accuracy.

3) Accuracy Results: For sequences MSA and Walk1, ground truth consists of information on bounding boxes of true moving objects for each sequence frame. From the values of frame-based measures reported in Table I we can observe that most of the considered algorithms perform quite well on the MSA sequence, with SOBS algorithm reaching the highest accuracy values. For the more challenging Walk1 sequence, all the considered algorithms attain lower accuracy values, but still SOBS method performs slightly better than the other methods.

As already mentioned, for sequences belonging to [4], ground truth is available as a binary detection mask for one reference frame. In the case of sequence ToD, the very low illumination level and the consequent low contrast between the moving person and the background lead to pixel-based measure values not very high, as it can be observed in Table II. Here, SOBS attains accuracy values generally higher than all the other methods. Foreground masks obtained by all the considered algorithms are reported in Fig. 4. For sequence WT, Pfinder, as expected, cannot capture moving background, while all the others perform quite well, as reported in Table III. Specifically, SOBS performs generally slightly better than all the other methods. Foreground masks obtained by all the considered algorithms are shown in Fig. 5.

Finally, for sequence IR, ground truth is available as a binary detection mask for 113 reference frames. Average pixel-based accuracy values over all reference images are reported in Table IV, where it can be observed that SOBS attains accuracy

TABLE IV
AVERAGE PIXEL-BASED ACCURACY VALUES FOR SEQUENCE IR

	Recall	Precision	F_1	Similarity
SOBS	0.84	0.95	0.89	0.80
Pfinder	0.82	0.86	0.83	0.72
VSAM	0.81	0.91	0.85	0.75
CB	0.75	0.93	0.83	0.71

(a) (b) (c)

Fig. 7. Segmentation of sequence CAM. (a) Original frame. (b) Ground truth. (c) SOBS result.

values higher than all the other methods. Foreground masks obtained by all the considered algorithms on one of the reference frames are shown in Fig. 6.

4) Further Comparisons: Further comparisons have been made with background subtraction results reported in [23]. Specifically, the authors in [23] employ a feed-forward neural network to achieve background subtraction, named *BNN*, that is a combination of a probabilistic neural network and a winner-take-all neural network, with the addition of rules for temporal adaptation of the network weights based on a Bayesian formulation of the segmentation problem. They compare results achieved by *BNN* with those obtained by *mixture of Gaussian* (MoG) [12] and by the method of Li *et al.* [24].

The *MoG* method uses multiple Gaussian distributions as a model for the values of the background pixels and an on-line approximation to update the model. The Gaussian distributions are then evaluated to determine which are the most likely to result from a background process.

Li *et al.* [24] model the pdfs of the pixel values detected by a filter-based initial segmentation, in order 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.

For comparison, we have considered the image sequences adopted in [24] and publicly available at http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html. For all the sequences, hand-segmented background is given for twenty test frames randomly chosen along the sequence.

Qualitative results in terms of foreground masks obtained by SOBS algorithm on one of the test frames for each sequence are shown in Figs. 7–15 and can be readily compared with those obtained by *BNN*, *MoG*, and Li *et al.* reported in [23] (not included here). SOBS algorithm parameter values that are not common to all the other reported experiments are detailed in Table V. Values for K, ϵ_1, and ϵ_2 have been chosen according to considerations reported in Section IV-A, while values for c_2 are set as learning rates adopted in [23], where the choice has been based on the observed speed of foreground objects motion and the background rate of change for each sequence.

Quantitative results reported in [23], expressed in terms of average *Similarity* values over all test images for each sequence,

(a) (b) (c)

Fig. 8. Segmentation of sequence FT. (a) Original frame. (b) Ground truth. (c) SOBS result.

(a) (b) (c)

Fig. 9. Segmentation of sequence WS. (a) Original frame. (b) Ground truth. (c) SOBS result.

(a) (b) (c)

Fig. 10. Segmentation of sequence MR. (a) Original frame. (b) Ground truth. (c) SOBS result.

(a) (b) (c)

Fig. 11. Segmentation of sequence LB. (a) Original frame. (b) Ground truth. (c) SOBS result.

(a) (b) (c)

Fig. 12. Segmentation of sequence SC. (a) Original frame. (b) Ground truth. (c) SOBS result.

(a) (b) (c)

Fig. 13. Segmentation of sequence AP. (a) Original frame. (b) Ground truth. (c) SOBS result.

(a) (b) (c)

Fig. 14. Segmentation of sequence BR. (a) Original frame. (b) Ground truth. (c) SOBS result.

(a) (b) (c)

Fig. 15. Segmentation of sequence SS. (a) Original frame. (b) Ground truth. (c) SOBS result.

TABLE V
SOBS ALGORITHM PARAMETERS VALUES ADOPTED FOR RESULTS SHOWN IN FIGS. 7–15

	CAM	FT	WS	MR	LB	SC	AP	BR	SS
K	100	100	100	200	100	300	600	800	300
ϵ_1	0.1	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1
ϵ_2	0.01	0.003	0.005	0.005	0.001	0.001	0.005	0.001	0.007
c_2	0.05	0.01	0.01	0.003	0.01	0.01	0.03	0.03	0.05

TABLE VI
COMPARISON OF SIMILARITY VALUES OBTAINED BY *BNN*, *MoG*, AND LI *ET AL.* REPORTED IN [23] WITH THOSE OBTAINED BY *SOBS* FOR SEQUENCES PROVIDED IN [24]

	CAM	FT	WS	MR	LB
SOBS	0.6960	0.6554	0.8247	0.8178	0.6489
BNN	0.5256	0.4636	0.7540	0.7368	0.6276
MoG	0.0757	0.6854	0.7948	0.7580	0.6519
Li et al.	0.1596	0.0999	0.0667	0.1841	0.1554

	SC	AP	BR	SS	
SOBS	0.6677	0.5943	0.6019	0.5770	
BNN	0.5696	0.3923	0.4779	0.4928	
MoG	0.5363	0.3335	0.3838	0.1388	
Li et al.	0.5209	0.1135	0.3079	0.1294	

are compared in Table VI with those obtained using the proposed SOBS algorithm. Here, we can observe that our approach generally achieves average *Similarity* values fairly higher than those obtained by BNN, MoG, and the method proposed by Li et el. for almost all the sequences.

From qualitative and quantitative results, we conclude that SOBS was successful in coping with complex background variations, such as waving trees in sequence CAM (Fig. 5), water fountain in sequence FT (Fig. 8), sea waves in sequence WS (Fig. 9), moving curtain in sequence MR (Fig. 10), and moving escalator in sequence SS (Fig. 15).

Moreover, SOBS succeeded in including shadows into the background model, as it can be observed by masks produced for sequences LB, SC, AP, and BR (Figs. 11–14), where shadows cast by moving objects are particularly evident.

Also, bootstrapping problems related to sequences SC, AP, BR, and SS (Figs. 12–15), that always contain foreground moving objects, have been satisfactorily handled by SOBS algorithm with suitable choices of calibration parameter values for K and ε_1, leading to adequate initial background models (not reported here for space constraints).

TABLE VII
PERFORMANCE VALUES (IN FPS) FOR SEQUENCES MSA, WALK1, TOD, WT, AND IR

	MSA	Walk1	ToD	WT	IR
SOBS	31.17	22.28	99.90	97.75	32.01
Pfinder	56.43	35.59	151.29	148.81	57.20
VSAM	48.85	30.84	135.50	133.37	48.12
CB	25.83	17.59	85.54	84.82	25.51

Even though SOBS average *Similarity* values reported in Table VI are generally higher than those achieved with the other methods, for two sequences, namely FT and LB, they are still quite low. In the case of sequence FT, this is due to the presence of stationary objects during the whole calibration phase, such as the shoulder of a man into the lower right corner of the scene. In the case of sequence LB, low *Similarity* value is due to strong illumination changes. Indeed, the light in the lobby is shut down, causing the entire scene to become very dark. When this happens, most of the image is not modeled anymore by the background model and, being detected as moving object, is included into the foreground mask. Due to the selective update in the SOBS algorithm, pixels belonging to such mask are not updated into the corresponding background model, that, therefore, does not adapt to the new illumination conditions. This is not evident from detection result reported in Fig. 11, that has been obtained for the test image related to a sequence frame where the light had been turned on again. Indeed, the detection mask appears quite accurate, since the illumination conditions allow to use the background model built before the light switch.

5) Performance Results: Computational complexity of SOBS algorithm, both in terms of space and time, is $O(n^2NM)$, where n^2 is the number of weight vectors used to model each pixel and $N \times M$ is the image dimension. To complete our analysis, in Table VII we report the mean number of fps over the whole video sequences described in Section IV-A for the SOBS algorithm and for methods described in Section IV-B1. Timings have been obtained by prototype implementations in C programming language on a Pentium 4 with 2.40 GHz and 512-MB RAM, running Windows XP operating system, and do not include I/O. The table shows that SOBS improves performance speed of CB, but is always slower than Pfinder and VSAM. Only for higher resolution sequences (such as Walk1), the frame rate is not sufficient to obtain real-time processing. Some optimization of SOBS could be, for instance, in terms of pruning of the not winning weight vectors, although experimental results in this direction have not yet reported appreciable improvements.

V. CONCLUSION AND ONGOING WORK

We have presented a new self-organizing method for modeling background by learning motion patterns and so allowing foreground/background separation for scenes from stationary

cameras, strongly required in video surveillance systems. Unlike existing methods that use individual flow vectors as inputs, our method learns background motion trajectories in a self organizing manner; this makes the neural network structure much simpler.

This paper also includes a comprehensive accuracy testing, performed with both pixel-based and frame-based metrics, in the light of recent efforts towards the assessment of performance evaluation approaches for tracking and visual surveillance [1]. Experimental results, using different sets of data and comparing different methods, have demonstrated the effectiveness of the proposed approach, which proves also robust to moving backgrounds, gradual illumination changes, and cast shadows, and has no bootstrapping limitations.

The proposed method is inherently parallel, since computations for each pixel of each sequence frame can be done concurrently with no need for communications. This can help in lowering execution times for high-resolution sequences. Moreover, the approach is suitable to be adopted in a layered framework, where, operating at region-level, it can improve detection results allowing to more efficiently tackle the camouflage problem and to distinguish moving objects by those that, initially moving, have stopped. This is a very desirable operative mode, considering that a very actual visual surveillance task is looking for suspect abandoned luggage.

REFERENCES

[1] J. M. Ferryman, Ed., in *Proc. 9th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, 2006.

[2] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, "A system for video surveillance and monitoring," Tech. Rep. CMU-RI-TR-00-12, The Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, 2000.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 42–77, 1994.

[4] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proc. 7th IEEE Conf. Computer Vision*, 1999, vol. 1, pp. 255–261.

[5] L. Maddalena and A. Petrosino, "A self-organizing approach to detection of moving patterns for real-time applications," in *Proc. 2nd Int. Symp. Brain, Vision, and Artificial Intelligence*, 2007, pp. 181–190, Lecture Notes Comput. Sci. 4729.

[6] S.-C. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Proc. EI-VCIP*, 2004, pp. 881–892.

[7] M. Piccardi, "Background subtraction techniques: a review," in *Proc. IEEE Int. Conf. Systems, Man, Cybernetics*, 2004, pp. 3099–3104.

[8] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[9] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, May 1997.

[10] A. Elgammal, D. Hanvood, and L. S. Davis, "Nonparametric model for background subtraction," in *Proc. ECCV*, 2000, pp. 751–767.

[11] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. S. Davis, "Real-time foreground-background segmentation using codebook Model," *Real-Time Imag.*, vol. 11, pp. 172–185, 2005.

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

[13] R. Cucchiara, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1–6, Oct. 2003.

[14] B. P. L. Lo and S. A. Velastin, "Automatic congestion detection system for underground platforms," in *Proc. ISIMP*, 2001, pp. 158–161.

[15] G. Backer, B. Mertsching, and M. Bollmann, "Data- and model-driven gaze control for an active-vision system," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 12, pp. 1415–1429, Dec. 2001.

[16] V. Cantoni, M. Marinaro, and A. Petrosino, Eds., *Visual Attention Mechanisms*. New York: Kluwer, 2002.

[17] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[18] T. Kohonen, *Self-Organization and Associative Memory*, 2nd ed. Berlin, Germany: Springer-Verlag, 1988.

[19] R. B. Fisher, Change Detection in Color Images 1999 [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/PAPERS/iccv99.pdf

[20] M. Sarifuddin and R. Missaoui, "A new perceptually uniform color space with associated color similarity measure for content-based image and video retrieval," in *Proc. ACM SIGIR Workshop on Multimedia Information Retrieval*, 2005, pp. 1–8.

[21] J. R. Smith and S.-F. Chang, "VisualSEEk: a fully automated content-based image query system," in *Proc. 4th ACM Int. Conf. Multimedia*, 1997, pp. 87–98.

[22] EC Funded CAVIAR Project, IST 2001 37540 [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

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

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

Lucia Maddalena received the Laurea degree (cum laude) in mathematics and the Ph.D. degree in applied mathematics and computer science from the University of Naples Federico II, Naples, Italy.

She is a Staff Researcher at the Institute for High-Performance Computing and Networking, National Research Council of Italy. Her research interests include image processing, multimedia systems, and parallel computing.

Alfredo Petrosino (SM'02) is an Associate Professor of computer science at the University of Naples Parthenope, Naples, Italy. His research interests include image processing, pattern recognition, neural networks, and multimedia systems. He is the author of more than 70 referred papers and co-editor of five international volumes.

Prof. Petrosino is a Member of IAPR.