Abstract

In the last years, smart surveillance has been one of the most active research topics in computer vision because of the wide spectrum of promising applications. Its main point is about the use of automatic video analysis technologies for surveillance purposes. In general, a processing framework for smart surveillance consists of a preliminary motion detection step in combination with high level reasoning that allows automatic understanding of evolutions of observed scenes. In this paper, we propose a surveillance framework based on a set of reliable visual algorithms that perform different tasks: a motion analysis approach that segments foreground regions is followed by three procedures, that perform object tracking, homographic transformations and edge matching, in order to achieve the real-time monitoring of forbidden areas and the detection of abandoned or removed objects. Several experiments have been performed on different real image sequences acquired from a Messapic museum (indoor context) and the nearby archaeological site (outdoor context) to demonstrate the effectiveness and the flexibility of the proposed approach.

1 Introduction

Smart surveillance has the main task of using automatic video analysis technologies in video surveillance applications. The aim is to develop intelligent visual equipment to replace the traditional vision-based surveillance systems where human operators continuously monitor a set of CCTV screens for specific event detection. This is not only a quite tedious activity, but with increased demands of large coverage areas, continuous monitoring tasks quickly become unfeasible due to information overload for human operators.

In this paper, we propose a visual framework based on a motion analysis approach followed by high level logical algorithms to achieve automatic smart surveillance of both indoor and outdoor contexts. In particular two problems have been addressed: the monitoring of forbidden areas and the detection of abandoned or removed objects.
Extensive tests in a real context have demonstrated that the system detects in real time unexpected events and sends alarms to human operators containing the information related to the detected anomaly (access violation, removed object or abandoned object).

1.1 Related work

Current literature proposes different smart surveillance systems to measure traffic flow, monitor security-sensitive areas such as banks, department stores and parking lots, detect pedestrian congestion in public spaces, compile consumer demographics in shopping malls, etc. In Wu and Huang (1999), Cedras and Shah (1995), Gravila (1999), Aggarwal and Cai (1999), Hu et al. (2004) excellent surveys on this subject can be found and in Collins, Lipton et al. (2000) the most significant works presented before 2000 are collected.

Nearly every visual surveillance system involves a preliminary motion analysis step to segment regions corresponding to moving objects from the rest of an image sequence. In Nair and Clark (2002) a background subtraction approach based on a simple modelling procedure (the model was built by the average of observed pixel values), frequent updating and a fixed threshold was used to segment moving people in an office corridor. Blob features were then extracted from any moving person and supplied as input to an HMM (Hidden Markov Model) based classifier in order to recognize exiting, entering and break-in activities. A similar motion analysis approach was used in Boulay et al. (2003) to detect moving people in an image. Projections of segmented blobs were then learned and used to recognize people posture by metric comparison. In Haritaoglu et al. (2000) dynamic background subtraction using an advanced statistical updating model was used to detect foreground objects. Shape and periodic motion cues of foreground regions were then used to detect and track people (even when they are moving together, or interacting with each other) and to determine whether a person is carrying an object, so that it can be tracked during exchanges. In Wren et al. (1997), Remagnino et al. (2004) a statistical modelling of the background was achieved by a single Gaussian for each pixel (with continuous updating of mean and variance values). In Remagnino et al. (2004) the motion detection step was the basic step of a multi-agent architecture for the understanding of scene dynamics, merging
the information streamed by multiple cameras whereas in Wren et al. (1997) motion information was used to detect people in the scene and to obtain a 2D representation of their head and hands in a wide range of viewing conditions. In Stauffer and Grimson (2000) rather than explicitly modelling the values of all the pixels as one particular type of distribution, the authors modelled the values of a particular pixel as a mixture of Gaussians making in this way the surveillance systems adapt to deal robustly with lighting changes, repetitive motion of scene elements, tracking through cluttered regions, slow-moving objects, and the introduction or removal of objects from the scene. A compact representation of the foreground regions was used for both tracking moving blobs by Kalman filters and to recognize their activities (pedestrians on the path, pedestrians and lawn-mowers on the lawn, activity near the loading dock, cars, trucks, etc.) using an on-line Vector Quantization (VQ). The same approach to detect and track moving object was adopted in Dee and Hogg (2004) to monitor pedestrian behaviour in a car park and a foyer. In Collins et al. (2000) robust routines for detecting moving objects and tracking them through a video sequence using a combination of temporal differencing and template tracking were introduced. Detected objects were then classified into semantic categories such as human, human group, car, and truck using shape and colour analysis, and these labels were used to improve tracking using temporal consistency constraints. Further classification of human activity, such as walking and running, was also achieved. In Mittal and Davis (2003) authors used a Bayesian approach to classify each pixel as belonging to a particular person or the background. People were then modelled using colour information and tracked. Finally information coming from different cameras were combined to manage occlusion, and to localize people on the ground plane introducing a likelihood map of people positions. In Bobick and Davis (2001) motion information was collected to build a temporal template, i.e. a static vector-image where the vector value at each point is a function of the motion properties at the corresponding spatial location in an image sequence. The authors then developed a recognition method, matching temporal templates against stored instances of views of known actions. Low level image and video processing techniques needed to implement a modern visual based surveillance system have been
described in Foresti et al. (2005b). Change detection methods for both fixed and mobile cameras (pan and tilt) are introduced and the registration methods for multicamera systems with overlapping and non-overlapping views are discussed. Moreover, the problem of motion detection in presence of moving cameras has been further discussed by authors in Foresti et al. (2005a). Many remarkable works in the direction of motion segmentation and tracking avoiding background subtraction have been presented in Dalal et al. (2006), Wu and Nevatia (2007) and Ramanan et al. (2007).

All the aforesaid papers presented very interesting theoretical fundamentals but, unfortunately, they don’t face the challenging problem of developing a real surveillance system that could be used in real-world scenarios.

This critical subject is, however, discussed in a few papers: Bakhtari and Benhabib (2007), Huanga et al. (2008) and, in particular, in Kristensen et al. (2008), where the authors have included their algorithms for motion segmentation and tracking in a real embedded system for surveillance, including details about the hardware, such as the FPGA, and computational time. Finally an extensive treatment of the surveillance system tasks in realistic scenarios and related technical challenges has been proposed in Shah et al. (2007).

In this paper a sequence of reliable visual algorithms has been used to realize a surveillance framework that performs two main tasks: detection of forbidden area violation, and detection of abandoned and removed objects. In contrast to the aforementioned papers that have considered specific problems in limited experimental situations, we have devised the whole chain of our surveillance framework to afford the concrete problems of surveillance of indoor and outdoor contexts with varying lighting conditions, and different complexity of surrounding scenarios.

1.2 System Overview

In figure 1 the sequential steps that compose the proposed framework are reported: first of all, moving areas are detected by background subtraction and the shadows are, eventually, removed by using an approach based on a temporal photometric gain analysis. Each segmented object in the scene is then localized in a 3D reference system scene using
homographic transformations and it is tracked by using both appearance and motion based features. In the next step, the gradient images of the current real and segmented frames are generated by using the SUSAN operator (Smith, 1992). Finally, the results of the previous steps are analysed by a decision making algorithm that merges all the information to recognize different situations. The first step of the decision making algorithm checks the 3D positions of each segmented object in order to detect forbidden area violations. The second step of the decision making algorithm selects, from foreground areas, the static regions and then, by comparing the gradient images of the current real and segmented frames, it recognizes abandoned or removed objects.

The main ideas underlying our processing steps came from known algorithms presented in literature but we have introduced some modifications to consider concrete problematic situations of real contexts and to respect the processing constraints for a real application of the proposed surveillance framework. The main key points of the entire processing chain are:

- the moving object segmentation procedure uses an innovative approach based on a temporal analysis of neighbour pixel variations and limits the effects of lighting changes; differently from several literature papers (for example Stauffer and Grimson, 2000, Collins et al., 2000, Haritaoglu et al., 2000), variations in working conditions are immediately detected, and the relative output is always reliable, without blindness periods;

- the shadow removing approach achieves a precise object shape extraction; it makes use of a two-step procedure that, firstly, segments the foreground region into several sub-regions according to the uniformity of photometric gain, and then detects shadow points and removes corresponding regions entirely. In this way the main drawback of several similar work (for example Rosito, 2009, Yang, Lo et al., 2008), i.e. the inability to remove all shadow points but only the great part of them, is overcome;
the tracking procedure, by considering both the temporal state vector and the appearance features, achieves satisfactory performance but, at the same time, it is computationally lighter than previous works (for example Senior et al. 2006);

finally, the foreground object analysis goes beyond the limits of some related works (for example Ferrando et al., 2006 and Tian et al., 2005) and it is able to distinguish between abandoned and removed objects in a reliable way, independently from the background updating procedure.

Extensive experiments were performed in both indoor and outdoor contexts. In particular the proposed framework was tested on real image sequences acquired from the Messapic museum and the nearby archaeological site of Egnathia (south of Italy).

The Messapic museum of Egnathia is a building containing numerous, precious, fragile and non-renewable archaeological finds. Next to the museum there is an archaeological site, a large open area containing the remains of the Messapic town of Egnathia. Both the museum and the archaeological site are open to visitors and the traditional surveillance cameras are installed to avoid theft or damage. The videos from these cameras are usually monitored sparingly or not at all; in fact they are used merely as an archive, to refer back to in case of anomalous incidents. Surveillance cameras would be more useful if they could detect anomalies requiring the attention of a human operator as soon as these events were recognized. This work was intended for this scope.

The rest of the paper is organized as follows: Section 2 details the motion segmentation step; the algorithm for shadow removing is described in Section 3; the connectivity analysis with the following tracking steps are described in Sections 4 and 5. The final steps of high level interpretation of foreground object movements for the detection of forbidden area violations and removed/abandoned objects are explained in Sections 6 and 7. Section 8 reports experimental results in both indoor and outdoor environments and a discussion on computational factors.
2 Motion Detection

The motion detection procedure extracts the binary shapes of those objects on which the following algorithms have to work. It is composed of several steps: first of all, during a learning phase, a statistical background model is evaluated, as suggested in Collins et al. (2008). At the end of this step, at each time instant \( t \), a mean value \( B'(x, y) \) and a standard deviation \( \sigma_5(x, y) \) are evaluated for each point. Then, moving regions are detected by evaluating, for each point, both the radiometric similarity between consecutive frames and the radiometric similarity between the current image and the background model. During the whole process, an updating procedure is performed in order to adapt the background model to the lighting condition variations. The details of these steps will be described in the following subsections.
2.1 Motion Segmentation

The frame sequence is processed by a temporal image analysis procedure, which compares consecutive frames \( I^t(x, y) \) and \( I^{t-1}(x, y) \), as follows: for each pixel of the image \( I(x, y) \), a window \( W \) (centered around it), is built and the mean and variance intensity values \( m[W] \) and \( v[W] \) are evaluated. These values are compared with those obtained on the previous image of the sequence by evaluating the radiometric similarity as follows:

\[
R(I^t(x, y), I^{t-1}(x, y)) = \frac{m[W(I^t(x, y))] - m[W(I^{t-1}(x, y))]}{\sqrt{v[W(I^t(x, y))]} + v[W(I^{t-1}(x, y))]} \tag{1}
\]

The output of this operation is a new intermediate binary image \( I'_M \) built as follows:

\[
I'_M(x, y) = \begin{cases} 
1 & \text{if } R(I^t(x, y), I^{t-1}(x, y)) \leq \sigma_s(x, y) \\
0 & \text{if } R(I^t(x, y), I^{t-1}(x, y)) > \sigma_s(x, y) 
\end{cases} \tag{2}
\]

where \( \sigma_s(x, y) \) is the standard deviation automatically set during an initial learning phase, as suggested in Collins et al. (2008). In the image \( I'_M \), zero values correspond to static pixels whereas non zero values correspond to candidate moving pixels.

By using neighbouring points in a window to compare corresponding points, we have chosen to give a local interpretation at the concept of difference, rather than a pixel-based one (Fejes and Davis, 1997). In this way the algorithm become more robust against noise: the effect of a single noise pixel is limited by other pixels in the window \( W \). Moreover, thanks to this first evaluation performed just in the current image, effects of sudden light changes are smoothed, and resulting images are always reliable; it is a mandatory constraint for a system that has to work in real contexts, where no blindness periods are admitted. It is for this reason that pure background based algorithms, even if they could be perfect for other contexts (for example Stauffer and Grimson, 2000, Haritaoglu et al., 2000), do not fit exactly with this application context.
The window size has to be a reasonable trade-off between the ability to smooth agglomerates of noise pixels when large windows are used and the possibility to erroneously detect static points on the edge of moving objects. In our experiments we set a window size of 5x5 pixels.

However this temporal image analysis is not enough to extract the correct shapes of moving objects, because it does not avoid the presence of holes inside moving objects and discontinuities along their contours. For this reason, the radiometric similarity proposed in equation (1), is evaluated again between the current image (excluding the pixels corresponding to zero values in the image $I_M$) and a reference background image $B'$. Formally, the final foreground image $F'$ is built as follows:

$$F'(x, y) = \begin{cases} 
1 & \text{if } R((I'(x, y), B'(x, y)) < \sigma, (x, y) \lor I'_M(x, y) = 1 \\
0 & \text{if } R((I'(x, y), B'(x, y)) > \sigma, (x, y) \land I'_M(x, y) = 0 
\end{cases}$$

with zero values corresponding to static pixels and non-zero values corresponding to moving pixels.

2.2 Background updating

The output of the above motion detection procedure strongly depends on the reliability of the involved background image $B'$ and a reliable background model has to account for varying light conditions at each time instant. This imposes the updating of the reference background image on the basis of the background observed in each image of the sequence.

In this work we propose an updating procedure that updates all the pixels, even those covered by foreground objects. Traditional approaches, if in the scene there are objects that move slowly (a typical situation when people observe archaeological finds and paintings), do not modify the corresponding background pixels producing background models that are no more consistent with the reality. As a consequence, when objects move away, the background subtraction algorithms produce a large number of artefacts. The goal of the proposed updating approach is to avoid this situation, by maintaining a consistent reference
image even in correspondence of static foreground objects. The main idea underlying the proposed approach is that pixels with the same intensity value, even if they are far in the image, have the same lighting variations and then can be updated in the same way. For each pixel the photometric gain is evaluated as follows:

$$A^{t-1}(x,y) = \frac{I'(x,y)}{B^{t-1}(x,y)}$$

(4)

where $B^{t-1}(x,y)$ is the background model at time instant $t-1$, and $I'(x,y)$ is the current image. The photometric gains measured on pixels having the same intensity value $B^{t-1}(x,y) = b_i$ are used to evaluate a mean photometric gain as follows:

$$\mu(b_i) = \frac{1}{N(b_i)} \sum_{\{x,y\} \in I'} A^{t-1}(x,y)_{\{x,y\} \in I', B^{t-1}(x,y) = b_i}$$

(5)

where $\{b_i\}_{i=1..n}$ are the $n$ different intensity values that a pixel can assume (0..255), and $N(b_i)$ is the number of pixels in the background image $B^{t-1}(x,y)$ with intensity value $b_i$. The equation (5) is evaluated on all the possible intensity values of the image. At the end of this procedure, we have 256 mean values that represent the amount of the change of each intensity value (for visible pixel). These values are used to update all image points, according to the rule that pixels with similar intensity values changes in a similar way. The iterative background updating rule has been defined as follows:

$$B'(x,y) = B^{t-1}(x,y) \mu(B^{t-1}(x,y))$$

(6)

The proposed background updating procedure reveals a number of advantages: it reduces the effects of noise in the image (sudden variations of spot pixels are not included in the background model since they are averaged out by the behaviors of other pixels with the same intensity); it does not depend on the correct detection of static or moving points (all the pixels in the image are updated). When objects move slowly in the scene the corresponding background points are also updated reducing in this way the effects of ghost areas in the segmented image.
3 Shadow Removing

After the motion detection step, foreground pixels correspond not only to real moving objects in the scene but also to their shadows. Shadow pixels have to be removed because they alter the real shape of moving objects making inaccurate any further attempt to automatically understand scene contents for smart surveillance.

Shadow removing is an open challenge in computer vision community. Many proposed algorithms in literature (Cucchiara et al. 2003; Wang et al. 2006) try to solve this problem at a pixel level by analyzing the spectral content of each individual point with the undesirable resulting effect to remove just a few shadow points. Recently several ratio based approaches for shadow elimination have been proposed (Rosito, 2009, Yang, Lo et al., 2008); moreover, MRF-based spatial relationship has been recently used for shadow removing (Huang et al, 2006 and Wang et al. 2006). However, they usually suffer in removing all shadow points, instead of only some of them. For this reason we have combined the pixel ratio approach with a preprocessing segmentation procedure, with the goal of remove shadow regions instead of shadow points. Moreover, no post processing morphological operators (that could alter the real shape of objects if the removed points are quite numerous), as proposed in Liu et al (2006) are required.

Our procedure starts from the assumption that shadows are half-transparent regions which retain the same representation of the underlying background surface pattern. We try to detect moving regions that have a texture substantially unchanged with respect to the corresponding background regions.

The foreground image $F'$ is used as a mask to segment moving objects into small regions $\{F_i\}$ characterized by a constant (within a given threshold) photometric gain $\Lambda_i$; regions with the value of the photometric gain greater than one unit are considered as foreground regions, while those having photometric gain lower than one unit are considered as candidate shadow regions, and further processing is necessary. In particular, for each point of the candidate shadow regions $\{F_i\}$, the correlation value with neighboring points, belonging to the same sub-region, evaluated in the current image, is compared with the
corresponding one obtained at the same location on the reference background image. In this work, satisfactory trade-off with the computational time constraint has been reached by using a simplified version of this algorithm, that computes a simple ratio between adjacent points to estimate the correlation measure $D$ as follows:

$$D((x, y), (x', y')) = \frac{I(x, y)}{I(x', y')} - \frac{B(x, y)}{B(x', y')} \quad (7)$$

where $(x, y)$ and $(x', y')$ are two points belonging to the region $\{F_i\}$ randomly selected, but chosen in order to analyze in an iterative way all the region points. If $D$ is lower than an experimentally selected threshold (in our experiment $0.9$), the pixels $(x, y)$ and $(x', y')$ are strictly correlated, and they are labeled as shadow points.

Experimental results demonstrated that the shadow removing results obtained using only a two pixel ratio are similar to those obtained using a more complex correlation, but with reduced computational time. The two pixel ratio is a very fast shadow elimination algorithm, but in theory it could have the problem of removing not only the shadows but also some points of people whose texture is similar to the background model. However, in our experiments on different situations, the number of false positives over foreground regions is not relevant with respect to the number of correctly removed shadow points.

In figure 2 we can see the whole process of shadow removing. Firstly, the original image is presented (a); in (b) we can see the regions with a similar photometric gain: different grey level intensity have been used for different regions. It should be noted that real foreground object (a person in an outdoor environment) has been divided into different regions, according to the presence of different colors on the object (black pants, black and white shirt, skin, …), while just a region has been detected for the shadow. In (c) candidate shadow points have been marked, according to the two pixel ratio algorithm previously explained. Finally, in (d), we can see the output after the suppression of regions with a high number of candidate shadow points: just region relative to shadow has been removed, while regions relative to the foreground objects have been maintained.
Figure 2. Example of the shadow removing procedure. (a) The original image; (b) output of the region segmentation algorithm based on the uniformity of photometric gain; (c) shadow pixels detected with the two pixels ratio; (d) foreground after the elimination of regions with a great number of shadow points.

4 Connectivity Analysis

The following step aggregates pixels belonging to the same moving object in order to build a higher logical level entity. The proposed framework makes use of the 8 connectivity criterion (Gonzales and Woods, 1993): two pixels are part of the same object, regardless of whether they are connected along the horizontal, vertical, or diagonal direction.

Each object in the scene is then numerically labeled and its geometrical information as center of mass, bounding box and area are extracted. In particular the area value for each object in the scene is used to detect object overlapping events. Assuming that most of the time the object in the scene there are isolated human beings, their most probable area is automatically found by the system after a few frames containing moving objects and, in the following frames, moving objects with area much greater than the estimated value are considered the results of some overlap and they are not considered by the decision making procedure in order to avoid false alarm.

5 People Tracking

Many algorithms have been proposed in the related literature on the tracking of multiple interacting objects in complex scenes. However this problem is still far from being completely solved: most of proposed approaches are designed to deal with short duration and partial occlusion, they fail when a group situation lasts for a long time or with non-linear motion (Rosales and Sclaroff, 1998) or do not solve more than two blob merge/split
(Yang, Li et al. 2005). Recently, more complex appearance models, which place a high burden on the CPU (Senior et al. 2006) have been proposed to increase reliability.

In this paper a temporal state vector is used to model the object dynamics and it is combined with a color histogram based procedure: this assures good performance but, at the same time, it does not clash with the real time requirement of the proposed framework.

To define the problem of people tracking it is necessary to estimate the state vectors that describe the considered dynamic system and the measurement vectors that represents the observations related to the state vectors.

In order to manage the split and merge situations and to have always one blob associated with one person we defined the state of the \( i \)-th person by 

\[
x_i^t = (p^i_x, v^i_x, d^i_x, l_x^i, h_x^i, s_x^i)
\]

where

- \( p_x^i, v_x^i \), and \( d_x^i \) represent the BB position, velocity, and dimension respectively.
- \( s_x^i \) is the BB status. It assumes the values: 1 for a single blob in the image, 2 for a merge blob, 3 for an exiting blob, 4 for a disappeared blob, 5 for single blob belonging to a group blob.
- \( l_x^i \) is a single label if the blob is a single blob, or a set of labels if the blob is a merge blob.
- \( h_x^i \) is the blob color histogram if the blob is a single blob or a set of blobs color histograms if the blob is a merge blob.

We denote the multi people configuration at time \( t \) with \( X_t = \{x_i^t | i = 1 \ldots N_t\} \), where \( N_t \) is the number of predicted BB in the image. In the same way we describe the measurement vector \( Z_t = \{z_i^t | j = 1 \ldots M_t\} \) where \( z_i^t \) are the observation instance vectors

\[
z_i^t = (p_x^i, h_x^i, d_x^i)
\]

\( M_t \) is the number of BB observed at the time \( t \). The observations \( Z_t \) are the results of the segmentation step and blob colour histogram computation. They are independent of previous instances, but to avoid false blobs due to noise (after a background
updating many artefacts can be introduced in the segmented image) they have to be validated by successive observations. Therefore \( M_t \) is not the number of BB observed but actually the number of BB validated by consecutive and coherent temporal and spatial observations.

At each step we have to predict the new state configuration considering the past state evolution and then validate this prediction with the new measurements. Supposing a linear model \( (f) \) of the people motion we can predict the new state configuration as \( X_t = f(X_{t-1}) + N \) where \( N \) is gaussian noise.

In this new prediction according to the position and kinematic parameters of the moving blobs in the scene we can have:

- \( x^i \) is a single track, that is a position change of one previous blob, if there is an \( x^{i,j}_{t-1} \) whose predicted position is in the image, (the state \( s^i_t \) is 1);
- \( x^i \) is a merge, if there are two or more blobs \( (x^i_{t-1}, x^h_t, \ldots) \) whose predicted positions fall close in the image, (the state \( s^i_t \) is 2);
- \( x^i \) can be an outgoing blob if there is a blob \( x^{i,j}_{t-1} \) whose predicted position is outside the image, (the state \( s^i_t \) is 3);

In the case of prediction of merge blob, the previous instances \( (x^{i,j}_{t-1}, x^h_t, \ldots) \) that generated the group blob are still maintained in the prediction \( (x^i_t, x^h_t, \ldots) \) with state \( s^i_t \) and \( s^h_t \) equals to 5. The prediction is also carried out for the \( x^m_{t-1} \) that have the status of disappeared blob (equal to 4).

As soon as the new measurement \( Z_t \) is available at the time \( t \) the prediction \( X_t \) has to be validated. By comparing all the observations \( z^h_t \) with \( h \in M_t \) and the predictions \( x^i_t \), several situations may happen: 1) some observations are close to the predictions and have the same colour histograms; 2) there are some state predictions \( x^i_t \) that do not correspond to any observation and 3) there are some observations \( z^h_t \) that do not match any state
prediction. In the first case, the predictions are updated considering the information of the corresponding observations (in particular the position, velocity and dimension fields are estimated). In the second case, if the prediction $x_i$ has not a correspondent among the observations and it is not on the image border (it is not in a outgoing situation), it means that the foreground segmentation procedure was not detecting the blob and then the status vector is maintained setting $s_i = 4$ (disappeared blob).

In the third case different situations may occur: 1) the observation $z_i$ could be a new entry blob if its $p_i$ is on the image border, then a new prediction $x_i$ is generated with an incoming state; 2) the observation $z_i^h$ could be a resumed blob if it is close to a prediction with a disappeared state; 3) the observation $z_i^h$ could be generated by noise, then a new entity $x_i$ is created and observed along a temporal window until a decision on its persistency is taken. Further analysis is required for merge blobs. We can predict that two or more blobs will merge (we set the status equal to 2), but since we need to maintain their vector status separated we have to split them in the corresponding observation. This splitting procedure can be difficult especially when two or more persons are very close to each other and the occlusion is almost total. However, when a merge blob is detected by the tracking procedure, it also maintains the information about the color histogram of the grouped people and the labels identifying the single tracked blobs. Starting from this information, a splitting procedure evaluates the group blob and searches for subregions having the same colour features of those searched. The search starts from the positions predicted by the single blob vector status and is enlarged as soon as the algorithm finds the best match. At the end of this step the state vector of each segmented blob is maintained by updating its position in the merge blob and setting the status $s_i = 5$ (blob belonging to a merge blob). The maintenance of state vectors for solved blobs in merge blobs allows us to recognize splitting situations. In fact when a split occurs (a single blob at time $t-1$ is divided in two or more blobs at time $t$) we have two or more observations $z_i, z_i^h, z_i^k$ matching with a prediction $x_i$ having $s_i = 2$ (merge status) and with two or more predictions $x_i^m, x_i^n$,
$x'_i$ having $s''_i = 5$ $s'^i = 5$ $s'_i = 5$. According to the number of objects in the merge blob and the number of observations we discriminate between a simple split or a complex split in single blobs and merge blobs. The decision is taken considering for each observation blob dimension, colour features and the best correspondence with the prediction having the status equal to 5. In this way we assign to the single blob $x''_i$ the new status $s''_i = 1$ (single track) and we remove or modify the merge blob $x'_i$ with $s'_i = 2$ reducing its number of internal objects.

6 Detection of intrusion into Forbidden Area

This procedure consists of two steps: firstly the 3D localization of moving objects is obtained using an homographic transformation; then object positions on the ground plane are compared with those labeled as forbidden in the foregoing calibration procedure.

In the first step for each detected moving object a point $p$ is considered: the point $p$ is obtained from the interception of a vertical line crossing the center of the bounding box of the considered object and the lower side of the same bounding box.

To localize the point $p$ in the 3D scene an homographic transformation between the image plane and the ground plane is introduced. The relation between the generic point $p(kx_i, ky_i, k)$ belonging to the ground plane and its corresponding point $p(u_i, v_i, 1)$ in the image plane is:

$$p = Mp \Rightarrow \begin{bmatrix} kx_i \\ ky_i \\ k \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}$$

(8)

To find the position in the scene of the moving object detected in the image plane, the 9 unknown elements of the matrix $M$ have to be computed ($m_{33}$ can be set to 1 since this is an homogenous linear system). The $m_{ij}$ elements can be estimated by considering 4 pairs of points whose coordinates both in the ground reference system and in the image plane
reference system are known a priori. This camera calibration is done just once during the system installation, then the matrix $M$ is used to project the position of all the moving objects observed in the image plane. At this point the detection of forbidden area violation can be carried out by comparing the moving object positions with the area coordinates labeled as forbidden in an initial calibration procedure. If a match occurs the algorithm generates an alarm signal.

7 Removed or Abandoned Object Detection

In many video surveillance applications it is very important to distinguish between abandoned and removed objects. In many works presented in literature, these two issues are not distinguished, and they are dealt with in a similar way. So, detecting an abandoned/removed object becomes a tracking problem, with the aim of distinguishing moving people from static objects left/removed by people (see Connell et al. (2004); Spengler and Schiele (2003) for good reviews).

In Porikli et al. (2008) a pixel based method is presented that distinguishes temporarily static image regions from the longer term background and moving regions by analyzing multiple foregrounds of different learning rates. Other interesting approaches are described in Ferrando et al. (2006) and Tian et al. (2005). In Ferrando et al. (2006) the authors use the shape of the foreground objects as a mask to distinguish between removed and stolen objects. Differences between the colour content of the area corresponding to the foreground object and the background around it, are evaluated in both the background and the current image and compared in order to perform an accurate foreground analysis. The basic idea in Tian et al. (2005) is quite similar, but now image edges (against colour histograms) are evaluated in both background and current image (in correspondence of foreground mask) and compared in order to distinguish between abandoned and removed object. The main drawback of both these approaches is their dependence from the background image: if it is not perfect, with the presence of artefacts due to an incorrect updating procedure, the whole process can fail. Moreover, they seem to be heavily dependant on the characteristics of the experimental context (textured background, textured foreground objects, …).
**Figure 3.** An example of an abandoned object in the corridor of a laboratory: (a) background model, (b) current frame with a red rectangle around the detected object, (c) segmented image (d) edge points detected in the red rectangle of the current image, (e) edge points detected in the red rectangle of the segmented image.

**Figure 4.** An example of a removed object in a room of the laboratory: (a) model of background, (b) current frame with a blue rectangle around the region of removed object, (c) segmented image, (d) edge points detected in the blue rectangle of the current image, (e) edge points detected in the blue rectangle of the segmented image.

Our work is based on the algorithms proposed by Ferrando et al. (2006) and Tian et al. (2005); we observed that their main drawback was their sensitivity to the soundness of the
background model: an incorrectly updated background, or a background unreliable due to extremely changing light conditions, could produce inconsistent results. To overcome this drawback, the approach we propose works directly on the current image and the segmented one, bypassing the processing of the background image: in this way, effects of light conditions are handled and smoothed in the segmentation algorithm, as described in section 2. In the experimental results section we prove the reliability of the proposed algorithm by comparing it with the two approaches aforementioned.

The approach implemented starts from the segmented image at each frame. If a blob is considered as static for a certain period of time (we have chosen to consider a blob as static if its position does not change for 5 seconds, but this value is arbitrary and does not affect the algorithm), it is passed to the module for removed/abandoned discrimination. The tracking module described in section 5, even if quite simple, is suitable for this purpose. By analyzing the edges, the system is able to detect the type of static region as abandoned object (a static object left by a person) or removed object (a scene object that is moved). First of all, an edge operator is applied to the segmented binary image $F'$ to find the edges of the detected blob (see figures 3c-4c and 3e-4e). The same operator is applied to the current gray level image $I'$ (figures 3b-4b) obtaining the results represented in figures 3d-4d (an optimization of this algorithm can work towards limiting this operation only to a correspondence of a region around the object previously detected). To perform edge detection, we have used the Susan algorithm (Smith, 1992), which is very fast and has optimal detection performances.

Let $E_F'$ and $E_I'$ be the two binary images obtained by applying, at time $t$, the edge operator respectively to the segmented image $F'$ and the current gray level image. To detect abandoned or removed objects a matching procedure of the edge points in $E_F'$ and $E_I'$ is introduced that counts the number of edge points in $E_F'$ that have a correspondent edge

---

1 In these and following figures in the paper, images have been colored, after processing, for display reasons.
point in $E'_f$. A searching procedure around those points is introduced to avoid mistakes due to noise or small segmentation flaws. Let $C'_F(x, y) \subset E'_F(x, y)$ and $C'_I(x, y) \subset E'_I(x, y)$ be two windows of size $n$ around point $(x, y)$ respectively in $E'_F$ and $E'_I$.

The matching measurement $M'_{FI}$ will be:

$$M'_{FI} = \frac{1}{N} \sum_{(x, y) \in E'_F(x, y)} M'_{FI}(x, y)$$

where $N$ is the number of edge points in $C'_F$ and

$$M'_{FI}(x, y) = \begin{cases} 1 & \text{if } E'_F(x, y) = E'_I(x, y) = 1 \\ \delta & \text{if } C'_I(x, y) \neq \phi \\ 0 & \text{if } C'_I(x, y) = \phi \end{cases}$$

$\delta$ is a coefficient that varies across $[0,1]$: it is equal to 1 if, in the region $C'_I$ around the point $(x, y)$, the number of edge points is greater or equal to those in the region $C'_F$ and it is 0 if the region $C'_I$ is empty. Formally:

$$\delta = \max\left(\frac{\sum_{(x, y) \in C'_I(x, y)} E'_I(x, y)}{\sum_{(x, y) \in C'_F(x, y)} E'_F(x, y)}\right)$$

After this procedure, the value of $M'_{FI}$ is sent to an if-then decision making module that can be summarized as:

$$M'_{FI} > th_a \Rightarrow \text{abandoned}$$

$$M'_{FI} < th_r \Rightarrow \text{removed}$$

otherwise $\Rightarrow \text{ambiguous}$

In other words, if $M'_{FI}$ value is greater than a certain value $th_a$ experimentally selected, it means that the edges of the object extracted from the segmented image have correspondent edge points in the current grey level image and it is labelled as an abandoned object by the automatic system. Otherwise, if $M'_{FI}$ has a small value, typically less then a given threshold $th_r$, it means that the edges of the foreground region do not match with edge points in the current image, so it is labelled by the automatic system as an object of the background that has been removed. For values of $M'_{FI}$ between these two thresholds the system is not able to decide on the nature of the object.
8 Experimental Results

Extensive experiments were performed in both the Messapic Civic Museum (indoor context) and the archaeological site (outdoor context) of Egnathia (Brindisi, Italy). The museum has many rooms containing important specimens of the past: the smallest archeological finds are kept locked in proper showcases but the largest ones are exposed without protection. The areas around the unprotected finds are forbidden areas for visitors and are delimited with cords. Only a visual control can ensure that visitors don’t step over the cords in order to touch the finds or to see them in more detail. The museum is adjacent to an archaeological site, i.e. a wide open area containing the remains of the messapic town of Egnathia. The archeological site consists of pathway (where visitors are allowed to walk) and constructions that cannot be approached due to the possibility of collapse. The proposed framework was tested to detect forbidden entry into protected areas of both the museum and archaeological site and to recognize removed and abandoned objects in the monitored areas.

In our experiment IEEE 1394 cameras were placed in the main room of the museum and near to the major pathway of the archaeological site. The acquired images were sent to a laptop (Pentium III, 1200 MHz, RAM 512, HD 30 Gb) where the algorithms described in the previous sections were processed. The next subsections explain the detail of the experiments carried out.

8.1 Indoor Experiments

In figure 5a) a sample frame acquired by the camera placed in the main room of the museum is shown: the four red markers indicate the points of the ground plane chosen for the calibration phase to estimate the homographic projection parameters. In figure 5b) the plan of the acquired area is reported: the green and pink colors indicate allowed and forbidden areas respectively. The red points correspond to the four red markers in figure 4a. In these figures the reference coordinate systems for both the image plane and the ground plane are also pointed out; in the image plane the \((u,v)\) coordinates measure the positions in
pixels whereas in the ground plane the \((x,y)\) coordinates measure the positions in meters in the real world coordinate system.

**Figure 5.** a) a frame acquired by the camera where the 4 red markers indicate the points of the ground plane chosen to discover the parameters of the homographic projection. b) the plan of the acquired area: the green color indicates allowed areas whereas pink color indicates forbidden areas. Red point correspond to the red point in figure 1A. Each figure reports also the reference coordinate systems used in the experiment.

The room was monitored for about 3 hours (30 frame/sec) during the visiting hours: several visitors came into the room but none was inside the forbidden areas or touched the archeological finds. In this experimental phase no false positives were found, that is the system didn’t provide any alarm in an improper way.

After the closing time some people performed illegal behaviors in order to validate the capability of the system to automatically detect them. A set of 29 sequences were recorded collecting 15 forbidden area violations, 8 abandoned objects, and 6 removed ones. The illegal behaviors were always automatically detected and recognized by the system.

Misclassification of human behaviors did not occur even in non trivial conditions. In particular, during the experimental phase, different people entered the scene at the same time and the sunlight shone through the large window with continuous changes of illumination conditions. The procedure for abandoned and removed object recognition did not fail even considering that textures in the static areas of the scene were not uniform and, in theory, this could cause false detections.
In figure 6 the left column shows some frames extracted while a person is stepping over the cord into the forbidden area, the central column shows the relative images containing the moving points, and finally the right column shows the results obtained after shadow removing. The relative position of the moving person on the image plane and onto the ground plane are respectively reported in the left and right columns. By comparing the position of the moving person on the ground plane with the boundary lines of the forbidden area the decision making procedure detected that in the third and fourth rows the person was performing an illegal access and sent an alarm.

**Figure 6.** The left column shows some frames extracted while a person is stepping over the cord; the central column shows the corresponding people segmentation results, the right column shows the relative images containing only the moving objects after the shadow removing.
In figure 7 the main advantage of using the proposed approach to detect illegal entrance into forbidden areas is highlighted: the figure on the left shows a visitor who is behind the cord but seems very close to the displayed find due to the perspective projection onto the image plane. In this case every approach based only on motion detection would wrongly detect an access violation and send a false alarm. The proposed approach instead detects the real position of the visitor and is able to label this situation as normal. In figure 7 on the right the visitor is inside the limit of the forbidden area. In this case the position estimation indicates the access violation and an alarm could be provided. It should be noted that cords are always positioned at a secure distance from important objects and then, as shown, the rough proposed localization of the people in the scene is adequate for this purpose: however the localization performance could be improved by using the center of mass of persons or adding a module for activity/posture recognition but, in those cases, the relative computational load should be taken under consideration to preserve the real time processing of the acquired scene.

Figure 7. Two critical situations: on the left a visitor stays behind the limit of the forbidden area but he seems very close to the find due to the perspective projection onto the image plane; the proposed system avoids the error of perspective perception and classifies this as normal behavior. On the right a visitor is inside the limit of the forbidden area. In this case the position estimation indicates the access violation and an alarm can be provided.

In figure 8 and 9 the effects of tracking algorithm are presented. In figure 8 two persons (in the image on the left) are close to a forbidden area, and they are segmented as a unique region by the segmentation algorithm (in the central image). Note that the homographic
position of this unique region is in a legal area, while it can be seen that one of the two persons is over the cord. In figure c) the two persons are correctly separated by the tracking procedure, and the system is now able to detect their exact positions, verifying that one of them is in the forbidden area (red rectangle in figure on the right). In figure 9 a more complex situation is presented: three people are connected in the scene but the tracking algorithm is able to separate them since they entered as separated entities in the camera field of view.

**Figure 8.** An example of the results of tracking algorithm: on the left two visitors stay close to the limit of the forbidden area, but one of them is behind it, while the other one is over the cord. In the central image the segmentation algorithm detects just one blob that seems to be in the legal area. The tracking algorithm separates the two visitors allowing the system to recognize the exact position of them in the image on the right.

**Figure 9.** An example of the results of tracking algorithm with three people connected in the scene: the algorithm is able to separate the three blobs since the three persons entered separated into the camera field of view. In this case no alarm was provided since no one was inside the forbidden area.

In figure 10 the correct detection of a removed object is shown: the three images show a person approaching the finds and stealing a piece of an ancient vessel. In the second row the processed images are shown: notice that a red rectangle (red rectangles indicate removed
object whereas blue rectangles indicate abandoned objects) is positioned around the area
where the removed object has been recognized.

![Figure 10. An example of automatic detection of removed object](image)

### 8.2 Outdoor Experiments

In figure 11-a a sample frame acquired by the camera placed near to the major pathway
of the archaeological site is shown: as before, in the image are reported the four red markers
used to evaluate the homographic projection parameters.

In figure 11-b the map of the acquired area is reported: the green color indicates allowed
areas whereas pink color indicates forbidden areas. The red points correspond to the red
markers in figure 11-a. In these figures the reference coordinate systems for both the image
plane and the ground plane are also reported.
The main path was monitored for about 2 hours (30 frame/sec) during visiting hours. The system correctly detected that visitors were inside the pathway and no false positive occurred. Also in this experiment, after the closing time, some people walked inside the forbidden area and left some objects in the scene in order to validate the capability of the system to automatically detect these suspicious events. A set of 15 sequences of forbidden area violation were correctly processed by the system and also 5 sequences containing abandoned objects were recognized. No false positive were found. We can conclude that also in an outdoor context the system can properly function and that the presence of varying light conditions in a rather complex scene does not affect performance.

In figure 12 the left column shows some frames extracted while a person entered the forbidden area containing the remains of the messapic town of Egnathia, the central column shows the relative images containing the moving regions and finally the right column shows the results obtained after shadow removing. The relative positions of the moving regions on the image plane and on the ground plane are respectively reported in left and right columns.
Figure 12 points out the importance of the shadow removing step. In the central column foreground pixels are displayed: it is very difficult to recover the correct shape of a moving object due to the associated long shadow. In these conditions the position of a moving object on the ground plane cannot be properly computed by homographic projection and the decision about the possible intrusion into forbidden areas will probably fail causing a large number of false alarms. These problems are instead suppressed in right column where binary images after shadow removing are shown. Using the information in right column people position on the ground plane could be, firstly, computed by homographic projection and then compared with the boundary lines of the forbidden area in order to recognize
intrusions. The person performing an illegal access is detected and an alarm can be provided.

Finally, in figure 13, the correct detection of an abandoned object in an outdoor context is shown; in the image on the left an actor approaches an ancient wall and in the central and right images he leaves some waste on site. In the images in the second row the output of the system is shown: notice that in the right image a blue rectangle has been automatically positioned around the area where the discarded object is positioned.

![Figure 13. Detection of an abandoned object in outdoor context](image)

8.3 Foreground Object Analysis evaluation

In order to compare the reliability of the proposed algorithm to distinguish abandoned objects from removed ones, we compared it with the approaches proposed in Ferrando et al. (2006) and Tian et al. (2005). In the following we call them respectively *Histo* and *EdgesCount* approaches, while *EdgesMatch* is the label used for the proposed approach.

We have tested all algorithms on different sequences, acquired in different contexts. In table 1 the main characteristics of them have been summarized: for each sequence, we indicate its name, the kind of events (abandoned/removed object), the number of frames of the sequence (acquired at 10 fps) and the topics of the context: in particular we have
combined two different kinds of background (textured or homogeneous) with two different kinds of objects (regular or irregular shape).

<table>
<thead>
<tr>
<th>NAME</th>
<th>Event</th>
<th>Frame</th>
<th>Background</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBRO_Ab</td>
<td>Abandoned</td>
<td>871</td>
<td>Homogeneous</td>
<td>Regular shape</td>
</tr>
<tr>
<td>HBRO_Rem</td>
<td>Removed</td>
<td>920</td>
<td>Homogeneous</td>
<td>Regular shape</td>
</tr>
<tr>
<td>TBRO_Ab</td>
<td>Abandoned</td>
<td>793</td>
<td>Textured</td>
<td>Regular shape</td>
</tr>
<tr>
<td>TBRO_Rem</td>
<td>Removed</td>
<td>821</td>
<td>Textured</td>
<td>Regular shape</td>
</tr>
<tr>
<td>HBIO_Ab</td>
<td>Abandoned</td>
<td>781</td>
<td>Homogeneous</td>
<td>Irregular shape</td>
</tr>
<tr>
<td>HBIO_Rem</td>
<td>Removed</td>
<td>1323</td>
<td>Homogeneous</td>
<td>Irregular shape</td>
</tr>
<tr>
<td>TBIO_Ab</td>
<td>Abandoned</td>
<td>883</td>
<td>Textured</td>
<td>Irregular shape</td>
</tr>
<tr>
<td>TBIO_Rem</td>
<td>Removed</td>
<td>1231</td>
<td>Textured</td>
<td>Irregular shape</td>
</tr>
</tbody>
</table>

In the following, the details about the implementation of the algorithms are presented; different experiments have been carried out, with different values for thresholds, histogram dimension, edges operator. The setups here reported are those that gave the best performance for each algorithm.

**Histo:** 8-bin histograms, threshold \( th=0.7 \) (difference greater then \( th \) = abandoned; else removed);

**EdgesCount:** the main idea is that a static region is considered as an abandoned object if there are significantly more edges with respect to the background, or a removed one if there are less edges, while no decision is taken in presence of similar number of edges. The algorithm has been implemented in this way: we evaluated the ratio between the number of edges in the background and the same in the current image; if this ratio is in the range \([0.85 \text{ - } 1.15]\) then no decision is taken. Otherwise, we have an abandoned or removed object according to the absolute number of edges.

**EdgesMatch:** in our approach we have fixed the following threshold in equation 12: \( th_a = 0.7, \ th_r = 0.4 \).

In table 2 the results obtained on the test sequences are reported. From left to right, the first column indicates the sequence label, the second one reports the ground truth, i.e. the
number of frames in the considered sequence containing the removing or abandoning event and the last three columns the performance of the comparing approaches. The last row in the table 2 indicates the overall reliability of the three algorithms is reported. As we can note, the three approaches work well, but in different way: this is due to their intrinsic characteristics.

Table 2. The results obtained comparing the proposed approach (last column) with two classical one (third and fourth columns).

<table>
<thead>
<tr>
<th>Name</th>
<th>Ground truth</th>
<th>Histo</th>
<th>EdgesCount</th>
<th>EdgesMatch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R: 120</td>
<td>R: 120 – 59.11%</td>
<td>R: 7 – 2.45%</td>
<td>R: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I: 16 – 7.88%</td>
<td>I: 16 – 7.88%</td>
<td>I: 4 – 1.07%</td>
</tr>
<tr>
<td>HBRO_Rem</td>
<td>R: 187</td>
<td>A: 2 – 0.11%</td>
<td>A: 0</td>
<td>A: 4 – 2.14%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R: 185 – 99.89%</td>
<td>R: 167 – 89.30%</td>
<td>R: 183 – 97.86%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I: 20 – 10.70%</td>
<td>I: 31 – 12.45%</td>
<td>I: 0</td>
</tr>
<tr>
<td>TBRO_Ab</td>
<td>A: 249</td>
<td>A: 148 – 59.44%</td>
<td>A: 208 – 83.53%</td>
<td>A: 222 – 89.16%</td>
</tr>
<tr>
<td></td>
<td>R: 101</td>
<td>R: 141 – 95.27%</td>
<td>R: 10 – 4.02%</td>
<td>R: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R: 101 – 40.56%</td>
<td>I: 31 – 12.45%</td>
<td>I: 27 – 10.84%</td>
</tr>
<tr>
<td>TBRO_Rem</td>
<td>R: 148</td>
<td>A: 7 – 4.73%</td>
<td>A: 0</td>
<td>A: 2 – 1.35%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R: 141 – 95.27%</td>
<td>R: 134 – 90.54%</td>
<td>R: 144 – 97.30%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I: 14 – 9.46%</td>
<td>I: 14 – 9.46%</td>
<td>I: 2 – 1.35%</td>
</tr>
<tr>
<td>HBIO_Ab</td>
<td>A: 120</td>
<td>A: 88 – 73.33%</td>
<td>A: 72 – 60.0%</td>
<td>A: 113 – 94.17%</td>
</tr>
<tr>
<td></td>
<td>R: 32</td>
<td>R: 32 – 26.67%</td>
<td>R: 2 – 1.67%</td>
<td>R: 2 – 1.67%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I: 46 – 38.33%</td>
<td>I: 46 – 38.33%</td>
<td>I: 5 – 4.16%</td>
</tr>
<tr>
<td>HBIO_Rem</td>
<td>R: 505</td>
<td>A: 13 – 2.57%</td>
<td>A: 65 – 12.87%</td>
<td>A: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R: 492 – 97.43%</td>
<td>R: 401 – 79.40%</td>
<td>R: 502 – 99.41%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I: 39 – 7.72%</td>
<td>I: 39 – 7.72%</td>
<td>I: 3 – 0.59%</td>
</tr>
<tr>
<td>TBIO_Ab</td>
<td>A: 315</td>
<td>A: 204 – 64.76%</td>
<td>A: 270 – 85.71%</td>
<td>A: 275 – 87.30%</td>
</tr>
<tr>
<td></td>
<td>R: 111</td>
<td>R: 111 – 35.24%</td>
<td>R: 13 – 4.13%</td>
<td>R: 4 – 1.27%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I: 32 – 10.16%</td>
<td>I: 32 – 10.16%</td>
<td>I: 36 – 11.43%</td>
</tr>
<tr>
<td>TBIO_Rem</td>
<td>R: 280</td>
<td>A: 63 – 22.50%</td>
<td>A: 31 – 11.07%</td>
<td>A: 25 – 8.93%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R: 217 – 77.50%</td>
<td>R: 222 – 79.28%</td>
<td>R: 226 – 80.71%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I: 27 – 9.65%</td>
<td>I: 27 – 9.65%</td>
<td>I: 29 – 10.36%</td>
</tr>
<tr>
<td>Total</td>
<td>R: 2207</td>
<td>TP: 1558 – 77.63%</td>
<td>TP: 1654 – 82.41%</td>
<td>TP: 1864 – 92.87%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FN: 225 – 11.21%</td>
<td>FN: 225 – 11.21%</td>
<td>FN: 106 – 5.28</td>
</tr>
</tbody>
</table>

In particular, the Histo algorithm is reliable in presence of removed objects, while it seems suffer when an object is abandoned on the scene. This difference is not evident for the EdgesCount approach; on the other hand, the use of edges makes it more dependent from the quantity of textures in both the background and the foreground object, and makes the results strictly depending on the reliability of the reference model. The proposed
approach does not consider the background image, so it is substantially independent of background limiting in this way the number of misdetection as shown in table 2. In general, in many test sequences the best results are those obtained with the proposed approach; only in the sequence HBRO_Rem, the Histo approach performed better than EdgesMatch.

9 Computational Remarks

In this section some considerations about the computational cost of the proposed approach are discussed. In table 3 the average computational load of each algorithmic step on a Pentium III, 1200 Mhz, RAM 512, HD 30 Gb is reported. An average value was computed on a set of 1000 images (500 for indoor context and 500 for outdoor context) with different contents. The image size was 640x480 pixels. The implementation environment was Microsoft Visual Studio .NET 2003. Table 3 shows that, due to the iterative implementation, the algorithmic steps that require more time to be accomplished are connectivity and shadow removing. On the contrary, edge detection is very fast because it is performed only on the segmented static blobs that have to be evaluated in order to understand if they are related to removed or abandoned objects. Summing up, the system was able to process about 30 frames per second allowing real time analysis of the scene. However, considering that the computation load per frame is not a constant value but depends on the complexity of the observed scene, in order to avoid loss of information at some critical moment (for example when a sudden light change occurs), a circular FIFO buffer has been implemented to queue up to 1024 acquired frames. If the processor is not busy the acquired frames are immediately processed otherwise they are store in the FIFO buffer. When the CPU becomes available the first frame in the buffer is removed from the buffer and processed. This trick guarantees that all the frames are processed and all the events in the scene can be detected. The possibility that acquired frames fill the buffer was not considered, since the experiments carried out in Egnathia demonstrated that the maximum number of frames in the buffer was 40 (so the events were detected with a maximum delay of 1 second). In figure 14 a plot of the buffer content is shown. The maximum number of frames (the peak in the plot) in the buffer occurred when a sudden
change in the light condition was recorded due to a cloud that darkened the sunlight for a few seconds. This event produced large moving areas on which the shadow removing procedure was applied increasing the computational load.

Figure 14. Buffer latency during about 30 seconds of the experimental phase in outdoor context

Table 3. Detailed Computational Load of each step involved in the proposed smart surveillance system

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Computational Load (in Milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Detection</td>
<td></td>
</tr>
<tr>
<td>Subtraction</td>
<td>3.767</td>
</tr>
<tr>
<td>Updating</td>
<td>1.928</td>
</tr>
<tr>
<td>Connectivity</td>
<td>9.697</td>
</tr>
<tr>
<td>Shadow Removing</td>
<td>11.365</td>
</tr>
<tr>
<td>Edge Detection</td>
<td>1.9887</td>
</tr>
<tr>
<td>Ground Plane Localization</td>
<td>0.0455</td>
</tr>
<tr>
<td>Decision Making procedure 1 (Forbidden Areas)</td>
<td>0.0365</td>
</tr>
<tr>
<td>Decision Making procedure 2 (Removed/Abandoned Objects)</td>
<td>1.1245</td>
</tr>
<tr>
<td>Total</td>
<td>29.9522</td>
</tr>
</tbody>
</table>

10 Conclusions

In this paper a smart surveillance system able to automatically detect unexpected events in the scene has been presented. The system makes use of different innovative processing
steps in order to achieve automatic real-time monitoring of forbidden areas and the
detection of abandoned or removed objects. Experimental test were performed in both
indoor and outdoor contexts. In particular the proposed framework was tested on real image
sequences acquired in the messapic museum and the adjoining archaeological site of
Egnathia (south of Italy). We chose to implement a visual surveillance system to satisfy the
strict requirement of a monitoring system that does not take advantage of any manipulation
of the objects to be monitored (for example by MEMS inertial sensors) and any fixed
installation in the surrounding environments (for example photoelectric cells).

The experimental results were very encouraging considering that no false alarms were
sent and all the unexpected events were correctly detected and labeled in real-time.

11 Acknowledgments

This work was developed under MIUR grant (ref. D.M. n. 1105, 2/10/2002) “Tecnologie
Innovative e Sistemi Multisensoriali Intelligenti per la Tutela dei Beni Culturali”.

References

Understanding, 73 (3), 428-440

190-200

IEEE Transactions on Pattern Analysis and Machine Intelligence, 23 (3), 257–267

BOULAY, B., F. BREMOND and M. THONNAT (2003) Human Posture Recognition in Video Sequence,
Tracking and Surveillance, 23-29

vol.13 (2), 129-155

COLLINS, R.T., A.J. LIPTON and T. KANADE (2000), Special section on video Surveillance, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (8), 745-887


HUANGA, K., L. WANGA, T. TANA, S. MAYBANK (2008), A real-time object detecting and tracking system for outdoor night surveillance, Pattern Recognition, 41, 432-444


The authors

Marco Leo

Marco Leo received the engineering degree in computer science from the University of Salento, Lecce, Italy, in 2001. Since 2001, he has been a Researcher with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. His research interests include image processing, computer vision, pattern recognition, digital signal processing, feature extraction, and neural networks. He has authored or co-authored over 100 technical publications including national and international journals, conference proceedings and book chapters. He is also co-author of three international patents. He currently serves as reviewer for several international journals. He participated at scientific committees of international conferences and symposia and organized special tracks in computer vision and multimedia.
**Paolo Spagnolo**

Paolo Spagnolo received the engineering degree in computer science from the University of Salento, Lecce, Italy, in 2002. Since 2002, he has been a Researcher with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. His research interests include computer vision, image processing and analysis, pattern recognition, and classification. He is the author of several papers in international journals and co-author of three international patents.

**Tiziana D’Orazio**

Tiziana D’Orazio received the computer science degree from the University of Bari, Bari, Italy, in 1988. Since 1997, she has been a Researcher with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. Her current research interests include pattern recognition, video analysis, and computer vision for video surveillance, domotics, intelligent transportation systems, and quality control. She is the author of more than 100 technical papers and book chapters in refereed conferences and journals in the areas of robotics and computer vision.

**Pier Luigi Mazzeo**

Pier Luigi Mazzeo received the engineering degree in computer science from the University of Salento, Lecce, Italy, in 2001. Since 2001, he has been with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. He participated in a number of national projects on real-time defect detection systems for railway maintenance using image processing. His research interests include signal processing, image processing, and image understanding.

**Arcangelo Distante**

Arcangelo Distante received the computer science degree from the University of Bari, Bari, Italy, in 1976. He is the Director of the Institute of Intelligent Systems for Automation, and the Coordinator of the Robot Vision Group, Italian National Research
Council, Bari, Italy. His research interests include computer vision, pattern recognition, machine learning, neural computation, robot navigation and computational models for the real-time processing of space/temporal image sequences. He is the co-author of more than 350 scientific papers.