

Robust unattended and stolen object detection by fusing simple algorithms

Juan Carlos San Miguel, José M. Martínez

Grupo de Tratamiento de Imágenes

Escuela Politécnica Superior, Universidad Autónoma de Madrid, SPAIN

E-mail: Juancarlos.Sanmiguel@uam.es, JoseM.Martinez@uam.es

Abstract

In this paper a new approach for detecting unattended or stolen objects in surveillance video is proposed. It is based on the fusion of evidence provided by three simple detectors. As a first step, the moving regions in the scene are detected and tracked. Then, these regions are classified as static or dynamic objects and human or non-human objects. Finally, objects detected as static and non-human are analyzed with each detector. Data from these detectors are fused together to select the best detection hypotheses. Experimental results show that the fusion-based approach increases the detection reliability as compared to the detectors and performs considerably well across a variety of multiple scenarios operating at real-time.

1. Introduction

Nowadays, surveillance systems[1] have more demand, specially for its application in public areas, as airports, stations, subways, entrance to buildings and mass events. (e.g., sports, concerts). In order to deploy more robust surveillance systems, the monitoring personnel could be helped by providing to them automatic analysis and interpretation tools able to focus their attention when a dangerous or strange event takes place in the monitored area, as well as allowing to effectively recover (from the video archive) the part of the sequence of video relative to some particular events.

In this context, reliable detection of the behavior of moving objects is an important and relevant requirement, and, more specifically, the detection of unattended or stolen objects in surveillance video is a highly relevant issue. For example, a useful application of unattended object detection could be to detect unattended packages in a subway station.. For stolen object detection, an example is suggested in[2], where a museum monitoring is proposed showing the importance of the stolen event detection. Automatic recognition of unattended or stolen objects in crowded unconstrained contexts is a

challenging task. Issues related to occlusions (by moving or static objects), appearance variations (e.g., colour composition, shape) as people move relatively to the camera, lighting changes, background scene complexity and the interactions between objects should be taken into account.

In this paper, an approach to detect unattended or stolen objects in surveillance video based on the combination of three fast detectors is presented. The detectors are based on the shape and colour information of the static foreground regions analyzed. A complete video surveillance system has been developed, including modules for the candidate object detection task and the proposed module for stolen or unattended object discrimination. The modules devoted to candidate object detection detects static foreground regions by a foreground segmentation and object tracking process, before classifying each object as human or non-human. Finally, static and non-human objects are analyzed by the proposed fusion-based approach to discriminate between unattended or stolen objects.

This paper is structured as follows: section 2 gives a brief overview of previous related work, section 3 overviews the proposed system, and section 4 explains the unattended or stolen detection module. In section 5, experimental results are shown, while section 6 closes the paper with some conclusions.

2. Previous Related Work

For the unattended or stolen object detection task, the following topics are relevant:

- Moving foreground object detection
- Object classification

For the foreground detection task, many approaches based on background subtraction were proposed to detect foreground objects. Such methods differ mainly in the type of background model and in the procedure used to update the model. In [3], a simple change detection module is used. This module takes into account pixel differences between consecutive frames and the frame-by-frame decisions accumulation. In [4], a scheme based on chromaticity distortion is presented. This system is able to

detect shadows, highlights and foreground regions. In [5] the authors propose to use spectral, spatial and temporal features, incorporated in a Bayesian framework, to characterize the background appearance at each pixel. More recently, the Gaussian Mixture Model (GMM) has been used in many works to model the background [6][7][8][9]. The GMM method is becoming popular because performs background estimation handling illumination changes, periodical motions, slow moving objects, long term scene changes, and camera noises. A good review about background modeling methods can be found in [10].

The object classification task is important for two reasons: it improves the quality of foreground detection and allows to discriminate among different object types (people/not people, unattended/stolen,...). In the last years, research has been mostly focused on detecting stationary objects and trying to discriminate them between people and unattended objects. For example, in [3] a system that accumulates temporal foreground results and a neural network classifier are used to discriminate between people, lighting effects and unattended objects. In [4] a multiple-state model of an unattended package provides the ability to detect realistic unattended package events. In [7] a long-term logic is used to differentiate between unattended objects and stationary people, and is robust to temporary occlusion of potential unattended objects. In [11], a knowledge based approach is performed to detect unattended objects. This approach is based on accumulated knowledge about human and non-human objects from continuous object tracking and classification. In [8] the input video is processed at different frame rates producing shot-term and long-term backgrounds modeled with a GMM each one. It detects stationary objects without using an object tracking step. In [12] a tracking stage and k-nearest neighbor classifier is used to detect foreground blobs as belonging to the bag or non-bag class depending on their size and shape. The decision is based on the detection of sub-events that characterizes the activity of interest.

The systems described above do not distinguish if the changes in foreground mask are due to unattended or stolen objects. Traditionally, these two issues are dealt in a similar way. So, detecting an unattended object becomes a tracking problem, with the classification problem of distinguishing moving or static people from static objects.

Recently, few papers can be found in the literature that change the classic tracking and people detection problem to a pattern recognition problem applied to the static foreground regions. These approaches can discriminate between unattended or stolen objects. In [9] an edge-based method is used to decide the type of static region by analyzing the change in edge energy associated with the boundaries of the static region between the current frame

and the background image. Similarly in [13], a statistical modeling of the background is used to detect foreground regions and to eliminate object shadows. Finally, moving objects are classified as unattended or stolen by matching the boundaries of static foreground regions. In [2], candidate objects extracted from the surveillance video are classified as human or non-human and static or dynamic. Finally static and non-human objects are analyzed to discriminate between unattended and stolen objects. This analysis is performed by analyzing the colour histogram of the regions determined by the object foreground mask and its surroundings. This colour analysis provides useful information to detect unattended or stolen object detection.

In conclusion, the different techniques in the reviewed literature use either edge or colour information that are used for the unattended/stolen object discrimination task. Numerous studies have discussed the advantages of the use of fusion schemes for combining multiple information sources like [14]. In the work presented in this paper, we propose a first approach to the combination of both types of information.

Also from the reviewed literature, results are usually provided over too few examples and with a vague analysis of complexity of the scene. In particular, none of the reviewed papers deal with the influence of the complexity of the scene background. As the presence of edges, multiple textures and moving objects belonging to the background can reduce dramatically the performance of the commented approaches, in our work we will provide results grouped by background complexity, together with scene complexity.

3. System Overview

A complete system has been designed to evaluate the proposed fusion-based approach. It performs two real-time tasks: candidate foreground object detection and unattended or stolen object discrimination. This paper is focused in the latter task.

The system is composed by different functional modules that have a specific task (see Figure 1). They are:

Foreground segmentation. This module does the foreground object extraction task. Foreground extraction is performed using an approach similar to the one of the semantic analysis module in [15] and a background update model based on a mixture between Average and Running Average methods [16]. The main advantage of this technique is that it can compensate a video signal with a time-varying noise level. Noise reduction of the binary foreground mask is performed using mathematical morphology. The operation is called opening by reconstruction of erosion [17]) and it preserves the underlying shape of the object. Finally blob extraction is performed on the binary foreground mask.

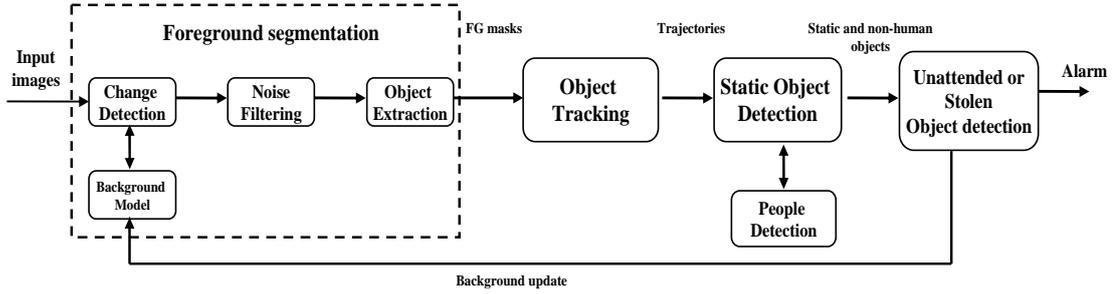


Figure 1: Overall System Architecture

Object Tracking. This module makes a simple tracking of the relevant objects identified in the previous module. It identifies different objects in the current analyzed frame and tries to associate them with the objects detected in the previous frame. This module does this work following different rules such as distance, colour, and object size.

Static Object Detection. This module has two main tasks for each foreground region: stationary blob detection and people confidence evaluation. The stationary detection is performed by constructing a trajectory graph for each blob and analyzing it. In this system, we have chosen to consider a foreground region as static if its position does not change during 50 consecutive frames. The people confidence evaluation is performed by using the *People Detection* module.

People Detection. This module calculates the confidence of being people for each blob. This module is structured as described in [18]. It is based on a fusion approach that combines the data provided by simple people detectors.

Unattended/Stolen Object Detection. This module is in charge of the unattended or stolen object detection task. It is performed by analyzing each “static” and “non-people” object. This module is described with more detail in section 4.

4. Unattended and stolen object detection

The unattended and stolen object detector proposed in this paper is based on the fusion of evidences derived from three fast detectors that are independently applied. The proposed approach is designed to exploit different types of information (shape similarity, contour similarity, background similarity, ...) and to fuse them to increase the the detection process robustness.

Figure 2 shows the processing steps for the unattended or stolen object detection task. Firstly, a pre-processing step is done to adjust the shape extracted from the binary foreground mask to the real object shape. This adjustment is made by using active contours[19]. Then three independent detectors are applied to the candidate object. Each detector calculates two evidence values of the being

unattended and stolen object hypothesis. Finally, a fusion scheme is applied on the evidences derived from the three detectors. As a result of the fusion process, two confidence measures are calculated for the likelihood of the object of being unattended or stolen. Afterwards, the maximum a posteriori criterion is selected to decide if the object is unattended or stolen.

In order to generate a measure of evidence from a given detector-related feature x , the latter is assumed to approximately follow a normal distribution of mean μ and standard deviation σ . Both parameters are experimentally determined for every defined feature by considering a training set with images/sequences of objects in different scenarios. The evidence of the given feature is then defined as a real value between zero and one, the latter when the feature is equal to its associated mean, as shown in Eq. (1).

$$E_{\mu,\sigma}(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

4.1. Shape adjustment by Active contours

Active contours[19] are used to adjust the object shape extracted from the binary foreground mask to the real

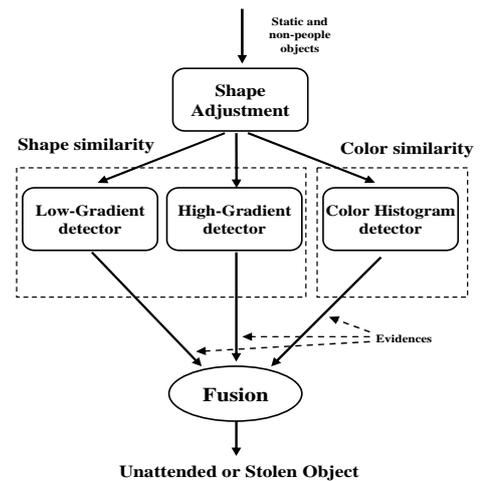


Figure 2: Flow processing for the event detection module.

object shape in the scene. This adjustment has to be done because estimation errors in the object shape can reduce dramatically the robustness of the algorithms used.

An Active contour is an energy minimizing function. It consists of a set of points that move within an image trying to find locations that will minimize their energy. Their energy is defined by the shape of the Active Contour itself and by its location in the image. The contour has two types of energy: Internal and External. The Internal Energy is defined by the shape of the Active Contour and is made up of Continuity Energy and Curvature Energy. Continuity Energy defines how the contour points are spaced from each other. Curvature Energy defines how smooth the contour is. The External Energy is the image energy. This is defined by the data within the image that the contour is attracted to. As we use the gradient image as the image energy, the data is attracted to the points, lines or curves of high gradient. This attraction to high gradients is why Active Contours are so suitable for object contour adjustment.

Specifically, the energy at every point of the Active Contour is defined as shown in Eq. (2).

$$E = \alpha E_{continuity} + \beta E_{curvature} + \lambda E_{image} \quad (2)$$

where α , β and λ are the parameters that are used to control the Active Contour.

Many different values of these three parameters were tested, and based on experimental results it was decided to use the following values: $\alpha=1.0$, $\beta=1.0$ and $\lambda=2.0$. A slightly higher value for λ is used to quickly converge to High-Gradient points. At each iteration every point of the Active Contour searches within a local neighborhood to find the new position that will minimize its energy. This local neighborhood is centered on the point's position at the start of that iteration. The size of the neighborhood was decided to set be to 3, so every point searches an area

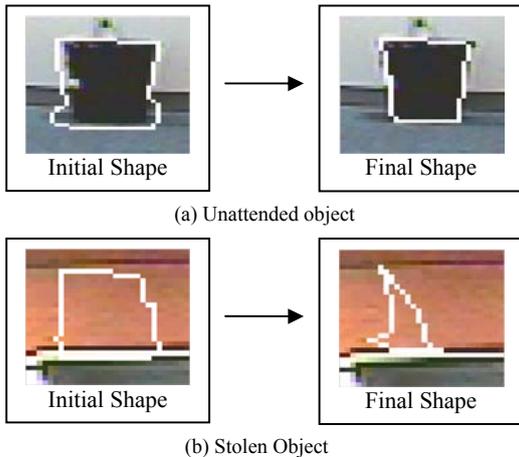


Figure 3: Active Contour adjustment example

of seven pixels squared at every iteration.

In Figure 3, two examples of the object shape adjustment are presented. These examples show how the shape is adjusted in case of unattended object presence (a) and deformed (not following the original object shape that produced the foreground region detection) when no object is present in the scene (stolen object case) (b).

4.2. Shape similarity detection

The first and second detectors to be combined are based on the shape similarity between the object shape adjusted by Active Contours and the real object shape in the current image (removing redundant shape information). This approach is similar to the one proposed in [13]. The main differences are the edge extraction process and the elimination of redundant edge information. In [13], the SUSAN algorithm is used to extract edges only providing binary information about them and being computational intensive. Some recent studies [20] show that the SUSAN algorithm is not the best choice for edge extraction. On the other hand, our approach extracts the edge information by using the gradient operator. This information extracted is richer than the one provided by the SUSAN algorithm (range 0-255 instead of binary range) allowing a better estimation of the edges (at pixel level). Some other properties (like edge connectivity or orientation) calculated by the SUSAN algorithm are not relevant for this application. After the edge extraction process, each detector calculates two confidence measures of being unattended or stolen object by analyzing the values of the gradient difference image between gradients of the current image and the background scene. This difference eliminates all gradient redundant information that is not useful for the discrimination task.

The first detector is based on finding the high-gradient value points in the current image that are along the object shape adjusted by Active Contours. The second detector is based on finding the low-gradient value areas in the same object shape used in the first detector. Due to the possibility that the gradient information extracted may not match exactly at pixel level, this search has to be generalized by using a small window around the examined point. In particular, the gradient matching measurement definitions are shown in Eqs. (3) and (4).

High-Gradient Matching

$$M_{HG}(x, y) = \begin{cases} 1 & \text{any}(W(x, y)) > th_{HG} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Low-Gradient Matching

$$M_{LG}(x, y) = \begin{cases} 1 & \frac{1}{N^2} \sum W(x, y) < th_{LG} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Background image	Current image	Object mask	Background image	Current image	Object mask
Gradient of background image for analyzed object	Gradient of current image for analyzed object	Gradient Difference between current and background image	Gradient of background image for analyzed object	Gradient of current image for analyzed object	Gradient Difference between current and background image
$M_{HG_U} = 0.70$		$M_{HG_S} = 0.30$		$M_{LG_U} = 0.32$	
(a)			(b)		

Figure 4: Example of object detection by High-Low gradient detector for unattended (a) or stolen (b) object for AVSS_AB_Medium/S1-T1-C-3 test sequences from i-LIDS Dataset for AVSS2007 and PETS06 dataset.

where $W(x, y)$ is the $N \times N$ window centered in the contour point (x, y) and th_{HG}/th_{LG} are the thresholds for High/Low Gradient detection (in this work we have heuristically selected the values $th_{HG}=220$ and $th_{LG}=30$ for the gradient detection task). For low-gradient detection we have decided to search value areas instead of finding value points. This is because edges are usually smaller than the window used, this fact implies that we can always find a low-gradient value in the observed window. The low-gradient measure would be wrong if we only try to search low-gradient value points. The proposed measure is more robust and is highly affected when a high-gradient value point is present in the observed window (reducing the false positives). Finally, all values of M_{HG} and M_{LG} are added and averaged by the total number of contour points to obtain two confidence values (for each detector) of the being unattended and stolen object hypothesis.

After this calculation, the values M_{HG} and M_{LG} are used to calculate the evidence of the contour similarity. For each class, unattended (U) and stolen (S), the evidence is defined as shown in Eq. (5) and Eq. (6).

$$E_{HG\{U,S\}} = E_{\mu_{HG\{U,S\}}, \sigma_{HG\{U,S\}}}(M_{HG}) \quad (5)$$

$$E_{LG\{U,S\}} = E_{\mu_{LG\{U,S\}}, \sigma_{LG\{U,S\}}}(M_{LG}) \quad (6)$$

where $\mu_{HG\{U,S\}}$, $\sigma_{HG\{U,S\}}$, $\mu_{LG\{U,S\}}$ and $\sigma_{LG\{U,S\}}$ are the mean and standard deviation of the High and Low gradient measures computed from the training set.

As we can see in Figure 4, the High-Gradient detector gives a measurement approaching to 1 when an object is unattended (case (a)) and to 0 for stolen objects (case (b)). On the other hand, the Low-Gradient detector gives a higher value (close to 1) for stolen objects (case (b)) and a

lower value (close to 0) for unattended objects (case (a)).

4.3. Colour similarity detection

The third detector to be fused is based on the colour similarity between the regions delimited by the object contour. This approach is similar to the one proposed in [2], where the similarity measure is proposed by checking whose change detection region in the current/background image (R2) is more similar to the corresponding surrounding region extracted in the current/background image and delimited by the bounding box and the shape of the object (R1 or "supposed background").

In our approach, we improve the colour measure proposed in [2] by using the same supposed background (calculated in the background image) for the colour differences. Assuming that the 'supposed background' is similar in the current and the background image is not always true. For example, in the case that two stolen events happen at the same time and neighboring regions, or that a unique stolen event is not properly detected due to oversegmentation of the region corresponding to the event, the measure proposed in [2] could reduce the performance depending on the new color information revealed behind the stolen objects. By using the same supposed background, the measure proposed is robust against this type of combined events. After the region extraction step, a colour histogram is calculated for the each of the three regions using the hue channel of the HSV color model (see Figure 5). Then the Battacharya distance[21] is calculated to find similarities between the colour histogram inside the object image (R2) in the current and background images (named H2 and H3 respectively), and the colour histogram outside the object image (R1) in the background image (H1). This algorithm

Background Image	Current Image	Object mask
		
Color histogram of R1 in background image (H1)	Color histogram of R2 in current image (H2)	Color histogram of R2 in background image (H3)
		

Figure 5: Example of object detection by Color information for S1-T1-C-3 test sequence from PETS06 dataset.

determines that the object is unattended if the histogram H1 is more similar to H3. On the other case, if the histogram H1 is more similar to H2, the object is considered to be stolen.

After this calculation, the value M_{CH} (calculated as the difference between the Battacharya distances of H1-H3 and H1-H2 histograms) is used to calculate the evidence of the colour similarity. For each class (Unattended and Stolen), the evidence is defined using Eq. (7).

$$E_{CH\{U,S\}} = E_{\mu_{CH\{U,S\}}, \sigma_{CH\{U,S\}}}(M_{CH}) \quad (7)$$

where $\mu_{CH\{U,S\}}$ and $\sigma_{CH\{U,S\}}$ are the respective mean and standard deviation of the difference distance for each class computed from the training set.

4.4. Data Fusion

The final evidences, E_U and E_S , about the analyzed object being unattended or stolen is obtained, in a first simple approach towards future more complex fusion schemes, by combining the evidences provided by the three detectors. The fusion scheme is defined as shown in Eq. (8) and Eq. (9). The evidence values are taken into consideration only if they are above the predefined relevance threshold ρ (heuristically we have selected $\rho = 0.7$ in this work).

$$E_U = \frac{H(E_{HGU} - \rho)E_{HGU} + H(E_{LGU} - \rho)E_{LGU} + H(E_{CHU} - \rho)E_{CHU}}{H(E_{HGU} - \rho)E_{HGU} + H(E_{LGU} - \rho)E_{LGU} + H(E_{CHU} - \rho)E_{CHU}} \quad (8)$$

$$E_S = \frac{H(E_{HGS} - \rho)E_{HGU} + H(E_{LGS} - \rho)E_{LGS} + H(E_{CHS} - \rho)E_{CHS}}{H(E_{HGS} - \rho)E_{HGS} + H(E_{LGS} - \rho)E_{LGS} + H(E_{CHS} - \rho)E_{CHS}} \quad (9)$$

where $H(x)$ is the Heaviside step function.

Finally, the candidate object is classified as stolen if $E_S > E_U$ and classified as unattended if $E_U > E_S$.

5. Experimental results

In this section, experimental results of the proposed method are presented. Experiments were carried out on several sequences from PETS2006 dataset (available at <http://www.pets2006.net/>), i-LIDS dataset for AVSS2007 and ChromaVSG[22] dataset. The system has been implemented in C++, using the OpenCV image processing library (www.intel.com/technology/computing/opencv/). Tests were executed on a Pentium IV with a CPU frequency of 2.8 GHz and 1GB RAM.

In order to evaluate the performance of the proposed approach, we have divided all test sequences into different complexity categories depending on the difficulty of the foreground object extraction and the background complexity. Foreground object extraction complexity is defined as the difficulty to extract stationary objects in a scenario. It is related with the number of objects, occlusions, lighting changes and the people detection difficulty. Background complexity is related with the presence of edges, multiple textures and objects belonging to the background (like waving trees, water surface,...). A description of complexity levels and length of the associated content is shown in Table 1

Table 1: Test sequences categorization.

Category	Length	Complexity	
		Foreground Extraction	Background
C1	19m 10s	Low	Low
C2	30m 05s	Low	Medium
C3	15m 10s	Low	High
C4	22m 35s	Medium	Medium
C5	25m 15s	High	Medium

A training set has been designed for the computation of the evidence models described in section 4 covering the different background complexities above described. This set is composed of around 10/10/5 minutes from the corresponding categories C1/C2/C3. The parameters of the models computed from the training set are: $\mu_{HGU}=0.75$, $\sigma_{HGU}=0.21$, $\mu_{LGU}=0.2$, $\sigma_{LGU}=0.16$, $\mu_{HGS}=0.17$, $\sigma_{HGS}=0.15$, $\mu_{LGS}=0.81$, $\sigma_{LGS}=0.14$ for the shape-based detectors and $\mu_{CHU}=-0.33$, $\sigma_{CHU}=0.18$, $\mu_{CHS}=0.17$ and $\sigma_{CHS}=0.092$ for the colour-based detector. The evaluation set is composed of the remaining sequences not used in the training set.

Due to space constraints, sample results are shown for the AVSS_AB_Medium test sequence extracted from the i-LIDS dataset for AVSS2007. Additional results can be

found at <http://www-gti.ii.uam.es/publications/AVSS08UattendedStolenObjectDetection>.

5.1. Overall system performance

Firstly, we present the evaluation of the performance of the overall system with test sequences. I should be noted that the same parameters were used for all the sequences, demonstrating the robustness of the proposed approach. The performance of the detection module is reported in terms of precision and recall. Precision indicates the percentage of alarms true and recall indicates the percentage of events detected. For the stationary object detection, we set the stationary object detection time to 2 seconds (50 frames at 25fps) and then the event detection process is started. Results from the sequences analyzed with the complete system have been summarized in Table 2. As can be seen from the results, our approach achieves better results in sequences with less foreground object extraction complexity and is robust to background complexity.

Table 2: Event Detection results

Scenario	Low Gradient		High Gradient		Colour Histogram		Fusion	
	R	P	R	P	R	P	R	P
C1	89%	75%	91%	76%	99%	80%	99%	80%
C2	86%	71%	88%	73%	90%	77%	93%	79%
C3	58%	50%	61%	55%	76%	72%	80%	75%
C4	74%	40%	70%	50%	75%	56%	76%	60%
C5	70%	25%	69%	28%	72%	32%	73%	34%

In Figure 6 we can see computational cost results for the overall system. The results shown for the High-Level modules are calculated as the mean of all candidate object processed (the time is negligible when there are no candidate objects). Some processing modules are independent of the number of objects of interest and other modules (like object tracking) add little computational

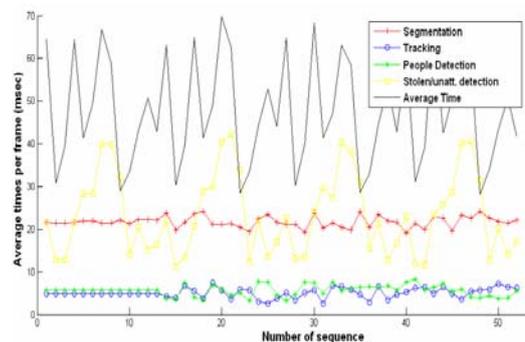


Figure 6: Processing times for the CIF sequences cost when the number of objects increases, due to their simplicity. On the other hand, High-Level processing modules are more dependent on the quantity and size of objects of interest present in each analyzed sequence. The real-time constraint limits the area of interest to analyze. For example, for CIF sequences at 20 fps the area is restricted to 15-18% of the image size.

5.2. Unattended/Stolen object detection evaluation

Secondly, the evaluation of the performance of the proposed fusion-based approach is presented. To evaluate it, we provide labeled data to the detection module, therefore decoupling the errors coming from previous modules from the ones inherent to the final module. Detection results for the individual methods and the fusion are shown in Table 3. These results are sorted in function of the sequence category analyzed and they demonstrate the capability and robustness of the proposed approach as it increases the success rate in all three categories.

Figure 7 illustrates an unattended detection example in the AVSS_AB_Medium test sequence from the i-LIDS dataset for AVSS2007. This example shows that although some detectors provide erroneous decisions (High-Gradient detector) the proposed fusion scheme determines a new decision selecting the evidences from the most confidence detectors and fusing them to generate a new

Gradient of background image for analyzed object	Gradient of current image for analyzed object	Gradient Difference between current and background image	Background Image	Current Image	Object Mask
Color histogram of R1 in background image (H1)	Color histogram of R2 in current image (H2)	Color histogram of R2 in background image (H3)			
Detector	Evidence		Decision		
	Unattended (U)	Stolen (S)	Method	Ground Truth	
Color Histogram	0.17	0.22	S	U	
High-Gradient	0.14	0.35	S	U	
Low-Gradient	0.85	0.0	U	U	
Fusion	0.85	0.0	U	U	

Figure 7: Unattended detection example for AVSS_AB_Medium test sequence from i-LIDS dataset for AVSS2007

decision improving the single detectors success rate for the unattended or stolen discrimination task.

Table 3: *True object detection comparison*

Background Complexity	True percentage detection (Precision)			
	Low Gradient	High Gradient	Colour Histogram	Fusion
Low	89.3%	91.43%	98.5%	99.7%
Medium	86.8%	90.9%	91.22%	93.58%
High	59.4%	60.4%	75.2%	76.4%

6. Conclusions

In this paper a new approach for real-time and robust unattended or stolen object detection is presented. It is based on the combination of the evidences generated by three independent detectors based on different extracted features. A first simple fusion scheme has been proposed, which combines only the most confident information sources. Experimental results show that this simple proposed scheme is significantly more efficient and stable than the independent detectors applied on their own. Experimental results also show that the main problem in the detection of this type of event is the stationary object extraction/detection.

7. Acknowledgments

This work is supported by Cátedra Infoglobal-UAM for “Nuevas Tecnologías de video aplicadas a la seguridad”, by the Spanish Government (TEC2007-65400 SemanticVideo), by the Comunidad de Madrid (S-050/TIC-0223 - ProMultiDis-CM), by the Consejería de Educación of the Comunidad de Madrid and by the European Social Fund.

8. References

- [1] K.N. Plataniotis, C.S. Regazzoni. “Visual-centric Surveillance Networks and Services”. IEEE Signal Processing Magazine, 22(2):12-15, 2005.
- [2] S. Ferrando, G.Gera and C. Regazzoni. “Classification of Unattended and Stolen Objects in Video-Surveillance System”. Proc. of IEEE Conf. on Video and Signal Based Surveillance, pp. 21-27, 2006.
- [3] C. Sacchi and C. Regazzoni. “A distributed surveillance system for detection of abandoned objects in unmanned railway environments”, IEEE Trans. On Vehicular Technology, 49(5):2013-2026, 2000.
- [4] M. Beynon et al. “Detecting abandoned packages in a multi-camera video surveillance system”, Proc. of IEEE Conf. on Advanced Video and Signal Based Surveillance, pp. 221–228, 2003.
- [5] L. Li et al. “Statistical Modeling of Complex Backgrounds for Foreground Object Detection”, IEEE Trans. on Image Processing, 13 (11):1459-1472, 2004.
- [6] C. Stauffer and W.E.L. Grimson. “Learning patterns of activity using real-time tracking”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(8):747-757, 2000.
- [7] N. Bird et al. “Real time, online detection of abandoned objects in public areas”, Proc. of IEEE Conf. on Robotics and Automation , pp. 3775-3780, 2006.
- [8] F. Porikli. “Detection of temporarily static regions by processing video at different frame rates”, Proc. of IEEE Conf. on Advanced Video and Signal Based Surveillance, pp. 236-241, 2007.
- [9] Y. Tian et al. “Robust and efficient foreground analysis for real-time video surveillance”, Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1182-1187, 2005.
- [10] M. Piccardi. “Background subtraction techniques: a review”, IEEE Trans. on System, Man and Cybernetics. (4):3099-3104, 2004.
- [11] S. Lu, J. Zhang and D. Feng. “A Knowledge-Based Approach for Detecting Unattended Packages in Surveillance Video”. Proc. of IEEE Conf. on Advanced Video and Signal Based Surveillance, pp. 110-115, 2006.
- [12] M. Bhargava et al. “Detection of abandoned objects in crowded environments”. Proc. of IEEE Conf. on Advanced Video and Signal Based Surveillance, pp. 271-276, 2007.
- [13] P. Spagnolo et al. “An Abandoned/Removed Objects Detection Algorithm and Its Evaluation on PETS Datasets”. Proc. of IEEE Conf. on Video and Signal Based Surveillance, pp. 17-21, 2006.
- [14] R. C. Luo and K. L. Su, “A review of high-level multisensor fusion: approaches and applications,” Proc. of IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems, pp. 25-31, 1999.
- [15] A. Cavallaro, A. Steiger and O. Ebrahimi. “Semantic Video Analysis for Adaptive Content Delivery and Automatic Description”, IEEE Trans. on Circuits and Systems for Video Technology, 15(10):1200-1209, 2005.
- [16] R. Cucchiara, et al. “Detecting Moving Objects, Ghosts, and Shadows in Video Streams”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(10):1337-1342, 2003.
- [17] P. Salembier and J. Ruiz. “On filters by reconstruction for size and motion simplification”, Proc. of Int. Symposium in Mathematical Morphology, pp.425–434, 2002.
- [18] V. Fernandez-Carbajales et al. “Robust People Detection by Fusion of Evidence from Multiple Methods”, Proc. of Int. Workshop on Image Analysis for Multimedia Interactive Services, pp. 55–58, 2008.
- [19] M. Kass, A. Witkin, and D. Terzopoulos. “Snakes: Active contour models”. International Journal of Computer Vision, 1(4): 321-331, 1988.
- [20] M. Shin et al, “Comparison of Edge Detection Algorithms Using a Structure from Motion Task”, IEEE Trans. on System, Man and Cybernetics, (31):589-601, 2001.
- [21] T. Kailath. “The Divergence and Bhattacharyya Distance Measures in Signal Selection”, IEEE Trans. on Communications Technology, 15(1):52-60, 1967.
- [22] F. Tiburzi et al, "A Ground-truth for motion-based video-object segmentation", Proc. of IEEE Int. Conf on Image Processing, (in press), 2008