NOVEL METRICS FOR PERFORMANCE EVALUATION OF OBJECT DETECTION ALGORITHMS *

Jacinto C. Nascimento Jorge S. Marques

Instituto de Sistemas e Robótica
Instituto Superior Técnico
1049-001 Lisboa
PORTUGAL

Abstract: This paper proposes novel metrics to evaluate the performance of object detection algorithms in video sequences. The proposed metrics allow to characterize the methods being used and classify the types of errors into region splitting, merging or merge-split, detection failures and false alarms. This methodology is applied to characterize the performance of five segmentation algorithms. These tests are performed in the context of object detection in outdoor scenes with a fixed camera.

Keywords: Performance evaluation, metrics, surveillance.

1. INTRODUCTION

Video surveillance systems rely on the ability to detect moving objects in the video stream. Each image is segmented by image analysis techniques. This should be done in a robust way in order to cope with unconstrained environments, non stationary background and different object motion patterns. Furthermore, different types of objects have to be considered e.g., persons, vehicles or groups of people.

Many algorithms have been proposed for object detection in video surveillance. They rely on different assumptions e.g., statistical models of the background (Wren *et al.*, 1997; Stauffer *et al.*, 2000) frame differences (Haritaoglu *et al.*, 1998) or a combination of both (Collins *et al.*, 1999). However, few information is available on the performance of these algorithms in different operating conditions.

Object detection assessment has been recently considered in (Oberti *et al.*, 1999) assuming it is a binary detection problem. Standard measures used in Communication theory such as misdetection rate, false alarm rate and receiver operating characteristics (ROC) were used (Trees, 2001). However, this approach has several limitations. Object detection is not a binary detection problem. Several types of errors should be considered (not just misdetections and false alarms). Second, the proposed test in (Oberti *et al.*, 1999) is based on the selection of rectangular regions with and without persons. This is an unrealistic assumption since practical algorithms have to segment the image into background and foreground and do not have to classify rectangular regions selected by the user.

In this paper, we propose objective metrics to evaluate the performance of object detection methods by comparing the output of the video detector with the ground truth obtained by manual edition of the video frames. The main features of the proposed method are the following. Given the correct segmentation of the video sequence we detect several types of errors $i)$ splits of foreground regions, $ii)$ merges of foreground regions, $iii)$ simultaneous split and merge of foreground regions, $iv)$ false alarms (detection of false objects), $v)$ the detection failures (missing active regions). They all influence the performance of the video surveillance system in different ways. Furthermore,

* This work was partially supported by FCT under project LTT and by EU project CAVIAR (IST-2001-37540). This work was previously published in PETS ECCV 2004.

ambiguous segmentations have also been explicitly considered. For example, it is not always possible to know if two close objects correspond to a single group or a pair of disjoint regions. Both interpretations are adopted in such cases.

Five segmentation algorithms are evaluated in this paper. The first is denoted as basic background subtraction (BBS) algorithm. It computes the absolute difference between the current image and a static background and compares each pixel to a threshold. All the connected components are computed and they are considered as active regions if their area exceeds a given threshold. The second method is the object detection algorithm proposed in the W4 system (Haritaoglu *et al.*, 2000). Three features are used to characterize each pixel of the background image: minimum intensity, maximum intensity and maximum absolute difference in consecutive frames. The third method assumes that each pixel of the background is a realization of a random variable with Gaussian distribution (SGM - Single Gaussian Model) (Wren *et al.*, 1997). The mean and covariance of the Gaussian distribution are independently estimated for each pixel. The fourth algorithm models each pixel as a mixture of Gaussians (Stauffer *et al.*, 2000), determining which mode corresponds to the background and which describe active regions (MGM - Multiple Gaussian Model). The fifth method is the one used in the *Lehigh Omnidirectional Tracking System* (LOTS (Boult *et al.*, 2001)).

The structure of the paper is as follows. Section 2 briefly reviews previous work. Section 3 describes the segmentation algorithms. Section 4 describes the performance metrics proposed in the paper. Experimental tests are discussed in section 5 and Section 6 presents the conclusions.

2. RELATED WORK

Surveillance and monitoring systems often require the segmentation of all the moving objects in the video sequence. Segmentation is a key step since it influences the performance of the other modules, e.g., object tracking, classification or recognition. For instance if object classification is required, an accurate detection is often needed to obtain a correct classification of the object.

Background subtraction is a simple approach to detect moving regions. This can be done in several ways. In (Haritaoglu *et al.*, 2000) each pixel of the background image is represented by three features: minimum intensity, maximum intensity, and the maximum rate of change at consecutive frames (Haritaoglu *et al.*, 2000), or the median of largest inter-frames absolute difference (Haritaoglu *et al.*, 1998). Background subtraction can be performed by combining different types of features. Other methods rely on the use of statistical models of the background image. The Pfinder ("Person Finder") (Wren *et al.*, 1997) assumes that the background is modeled by a Gaussian distribution: each pixel is described by its mean and covariance matrix. The object detection is based on blob detection. A blob is a connected region, obtained by clustering the pixels with similar color and image coordinates. A multiclass statistical model of color is used to monitor the activity of person in indoor environments. In (Stauffer *et al.*, 2000) segmentation is based on an adaptive background subtraction method which models each pixel as a mixture of Gaussians. In (McKenna *et al.*, 2000) it is proposed a background subtraction method which combines color and gradient information. Another approach is described in (Boult *et al.*, 2001). This system includes multiple background modelling, combining various techniques: adaptive thresholding with hysteresis and spatio-temporal grouping of active pixels, denoted quasi-connected components.

A problem related to the background subtraction approach are the false active regions, the so-called "negative" or ghosts (Siebel and Maybank, 2001). These regions are caused by static objects belonging to the background image (e.g., cars) which start to move. This gives rise to a false active region located where the object was placed, due to the difference between the current frame and the background model computed using past information. This problem can be overcome by high level techniques (Cucchiara *et al.*, 2002) or by modeling the background with a mixture of Gaussians (Stauffer *et al.*, 2000).

3. SEGMENTATION ALGORITHMS

This section describes object detection algorithms used in this work: *BBS, W4, SGM, MGM, LOTS*. The *BBS, SGM, MGM* algorithms use color while *W4* and *LOTS* use gray scale images. The BBS algorithm, detects moving objects by computing the difference between the current frame and the background image. A thresholding operation is performed to classify each pixel as foreground or background. Ideally, pixels associated with the same object should have the same label. This is accomplished by morphological filtering (dilation and erosion) to eliminate isolated pixels and small regions followed by a connected component analysis (e.g., using 8 - connectivity criterion).

The second algorithm is denoted here as $W4$ since it is used in the W4 system to compute moving objects (Haritaoglu *et al.*, 2000). This algorithm is designed for grayscale images. The background model is built using a training sequence without persons or vehicles. During this period three values are estimated for each pixel: minimum intensity (Min), maximum intensity (Max), and the maximum intensity difference between consecutive frames (D). Foreground objects are computed in four steps: $i)$ thresholding, $ii)$ noise cleaning by erosion, $iii)$ fast binary component analysis, $iv)$ elimination of small regions.

The thresholding step proposed herein is given by

$$(|I^t(x,y) < \mathrm{Min}(x,y)| \quad \vee \quad |I^t(x,y) > \mathrm{Max}(x,y)|) \\ \wedge\, |I^t(x,y) - I^{t-1}(x,y)| > D(x,y) \tag{1}$$

which leads to a less level of misdetections comparing with the one described in (Haritaoglu *et al.*, 2000). The third algorithm considered in this study is the SGM (Single Gaussian Model) algorithm. Color information is used in this method. Each pixel is represented by a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ where the mean μ and covariance Σ are recursively updated as follows

$$\boldsymbol{\mu}^t = \alpha I^t(x,y) + (1-\alpha)\boldsymbol{\mu}^{t-1}, \tag{2}$$

$$\Sigma^t = (1-\alpha)\Sigma^{t-1} + \alpha(I^t(x,y) - \boldsymbol{\mu}^t)(I^t(x,y) - \boldsymbol{\mu}^t)^T \tag{3}$$

The SGM performs a binary classification of the pixels into foreground or background and tries to cluster foreground pixels into blobs.

The fourth algorithm (MGM) models each pixel x = (x,y) as a mixture of three Gaussians distributions, i.e.

$$p(I(\mathrm{x})) = \sum_{k=1}^{N} \omega_k \mathcal{N}(I(\mathrm{x}), \boldsymbol{\mu}_k, \Sigma_k), \tag{4}$$

where $\mathcal{N}(I(\mathrm{x}), \boldsymbol{\mu}_k, \Sigma_k)$ is a multivariate normal distribution, $N = 3$ and ω_k is the weight of kth normal,

$$\mathcal{N}(I(\mathrm{x}), \boldsymbol{\mu}_k, \Sigma_k) = \\ c\ \exp\Big\{-\frac{1}{2}\Big(I(\mathrm{x}) - \boldsymbol{\mu}_k\Big)^T \Sigma_k^{-1}\Big(I(\mathrm{x}) - \boldsymbol{\mu}_k\Big)\Big\}. \tag{5}$$

with $c = \frac{1}{(2\pi)^{n/2}|\Sigma_k|^{\frac{1}{2}}}$. Note that each pixel $I(\mathrm{x})$ is a 3×1 vector with three component colors (red, green and blue), i.e., $I(\mathrm{x}) = [I(\mathrm{x})^R I(\mathrm{x})^G I(\mathrm{x})^B]^T$. To avoid an excessive computational cost, the covariance matrix is assumed to be diagonal (Stauffer *et al.*, 2000).

The mixture model is updated dynamically. $i)$ The algorithm checks if the pixel value x can be ascribed to a given mode of the mixture (match) [1] . $ii)$ If a distribution matches the new observation the parameters are updated according to (2), (3) where α is replaced by

$$\lambda_k = \alpha \mathcal{N}(I(\mathrm{x}), \boldsymbol{\mu}_k, \Sigma_k) \tag{6}$$

The weights in (4) are updated by

$$\omega_k^t = (1-\alpha)\omega_k^{t-1} + \alpha(M_k^t), \quad \text{with} \\ M_k^t = \begin{cases} 1 & \text{matched model} \\ 0 & \text{remaining models} \end{cases} \tag{7}$$

α is the learning rate. The non matched components of the mixture remain the same. If none of the existing components match the incoming observation, the least probable distribution is replaced with the current value (as its mean), a large covariance matrix and a low weight. This distribution should contain a high variance and mean equal to the current value of the frame, a low prior weight should be assigned in this situation. $iii)$ The distributions are sorted in the descending order of $\omega_k/|\Sigma_k|$. $iv)$ The algorithm selects the first B Gaussians distributions as belonging to the background. B is chosen as follows: B is the smallest integer such that

$$\sum_{k=1}^{B} \omega_k > T \tag{8}$$

where T is athreshold that accounts for a certain quantity of data that should belong to the background.

The fifth algorithm (Boult *et al.*, 2001) is tailored to the detection of non cooperative targets (e.g., snipers) under non stationary environments. This algorithm uses two gray level background images B_1, B_2. This allows the algorithm to cope with intensity variation due to noise or fluttering objects which move in the scene. Each pixel of the input frame is compared to the closest background value and classified as active if the difference exceeds a given threshold $T_L(x,y)$. A quasi connected component analysis is then performed using a second threshold $T_H(x,y)$ in order to select groups of active pixels and to classify them as targets. It is assumed that $T_H(x,y) = T_L(x,y) + c_S$, c_S being defined by the user.

These images are updated as follows

$$B_1(x,y) = \min\{I^n(x,y), n = 1, \ldots, T\} \tag{9}$$

$$B_2(x,y) = \max\{I^n(x,y), n = 1, \ldots, T\} \tag{10}$$

where $n \in 1, 2, \ldots, T$ denotes the different time instants during the adaptation period.

The background images are updated as follows. We compute the background image closest to the image intensity $I^t(x,y)$ and update it

$$B_i^{t+1}(x,y) = \begin{cases} (1-\alpha')B_i^t(x,y) + \alpha' I^t(x,y) & \text{if}(x,y) \in T \\ (1-\alpha)B_i^t(x,y) \ + \alpha I^t(x,y) & \text{if}(x,y) \in N \end{cases} \tag{11}$$

where α, α' are update gains $(\alpha' < \alpha)$, T is a target set, N is non-target set. α is usually small to enforce a slow adaptation. This is important to avoid the integration of active regions in the background image.

When updating the background image, the thresholds for pixels are also updated. Each pixel is first classified as false alarm, detection failure or target. Then the thresholds are updated

$$T_L^{t+1} = \begin{cases} T_L^t(x,y) + c_{FA} & \text{if false alarm} \\ T_L^t(x,y) - c_{DF} & \text{if detection failure} \\ T_L^t(x,y) & \text{if target} \end{cases} \tag{12}$$

In this paper we choose $c_{FA} = 10$, $c_{DF} = 1$. If the pixel is correctly classified as target the threshold remains the same. If the pixel is a false alarm the threshold is increased. If there is a detection failure the threshold is decreased.

[1] A match occurs if the pixel is inside the confidence interval with $+/-2.5$ standard deviation.

4. PROPOSED FRAMEWORK

In order to evaluate the performance of object detection algorithms we propose a procedure based on the following principles:

- A set of test sequences is selected. All moving objects are then detected and manually corrected if necessary to obtain the ground truth, one frame per second.
- The output of the automatic detector is compared with the ground truth.
- The output is then classified in one of the following classes: correct detection; false alarm; detection failure; merge; split; split-merge.

To perform the first step we made a user friendly interface which allows the user to define the foreground regions in the test sequence in a semi-automatic way (see the details in (J. Nascimento, 2004)).

The test images are used to evaluate the performance of object detection algorithms. In order to compare the output of the algorithm with the ground truth segmentation, a region matching procedure is adopted which allows to establish a correspondence between the detected objects and the ground truth. Several cases are considered:

(1) **Correct Detection (CD)** or **1-1 match**: the detected region matches one and only one region.
(2) **False Alarm (FA)**: the detected region has no correspondence.
(3) **Detection Failure (DF)**: the test region has no correspondence.
(4) **Merge Region (M)**: the detected region is associated to several test regions.
(5) **Split Region (S)**: the test region is associated to several detected regions.
(6) **Split-Merge Region (SM)**: when the conditions 4, 5 are simultaneously satisfied.

4.1 *Region Matching*

Object matching is performed by computing a binary correspondence matrix $\mathcal{C}$ which defines the correspondence between the active regions of a pair of images. Let us assume that we have N ground truth regions $\tilde{R}_i$ and M detected regions R_j at time t. Under these conditions $\mathcal{C}$ will be a $N \times M$ matrix, defined as follows

$$\mathcal{C}(i,j) = \begin{cases} 1 \text{ if } \tilde{R}_i \cap R_j \neq 0 \\ & \forall_{i \in \{1,\ldots,N\}, j \in \{1,\ldots,M\}} \\ 0 \text{ if } \tilde{R}_i \cup R_j = 0 \end{cases} \tag{13}$$

It is also useful to add the number of ones in each line or column, defining two auxiliary vectors

$$L(i) = \sum_{j=1}^{M} \mathcal{C}(i,j) \qquad i \in \{1,\ldots,N\} \tag{14}$$

$$C(j) = \sum_{i=1}^{N} \mathcal{C}(i,j) \qquad j \in \{1,\ldots,M\} \tag{15}$$

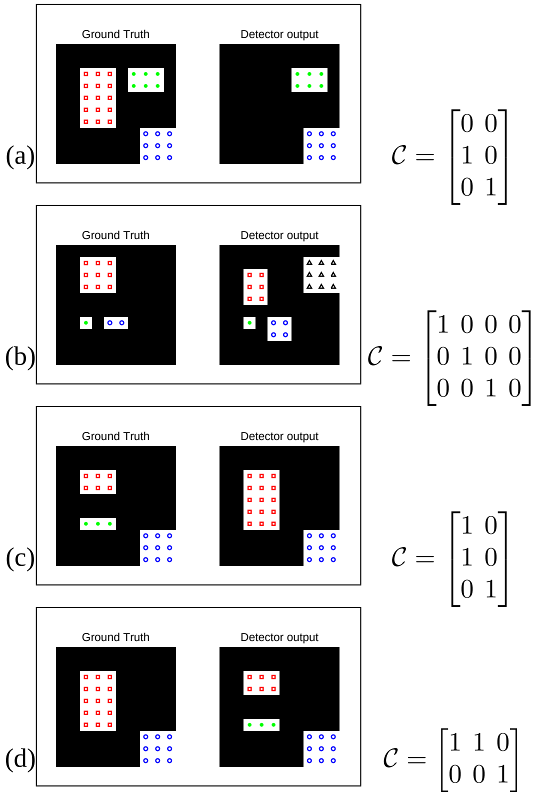
A match $\mathcal{C}(i,j) = 1$ can be classified as:

$$\begin{aligned} \text{correct detection: } & ifL(i) = C(j) = 1 \\ \text{merge: } & ifC(j) > 1 \\ \text{split: } & ifL(i) > 1 \\ \text{split-merge: } & ifL(i) > 1 \wedge C(j) > 1 \end{aligned} \tag{16}$$

The false alarms and detection failures can be formed detecting empty columns or lines in matrix $\mathcal{C}$. It is therefore easy to compute the statistics of each type of errors from matrix $\mathcal{C}$. Fig. 1 illustrates four situations considered in this analysis, by showing synthetic examples. Two images are shown in each case, corresponding to the ground truth (left side) and detected regions (right side). It is also depicted the correspondence matrix $\mathcal{C}$. For each case, the left image ($\tilde{I}^t$) contains the regions defined by the user (ground truth), the right image (I^t) contains the regions detected by the segmentation algorithm. Each region is represented by an white region containing a visual label. The first case in Fig. 1 (a) the "square-region" has no correspondence with the detected regions, thus it corresponds to a detection failure. In the following case (Fig. 1 (b)) the algorithm detects regions which do not match any region of I^t generating a false alarm. Fig. 1 (c) it is shown a merge of two regions since two different regions ("square" and "dot" regions in I^t) correspond to the "square region" in I^t. Finally, it is illustrated the split situation.

Sometimes the segmentation procedure is subjective, since each active region may contain several objects and it is not always easy to determine if it is a single connected region or several disjoint regions. For instance, Fig. 2(a) shows an input image with a manual segmentation. Three active regions were considered: person, lorry and group of people. Fig. 2 (b) shows the segmentation results provided by the SGM algorithm. This algorithm splits the group into three individuals which can also be considered as a valid solution since there is very little overlap. This segmentation should be considered as an alternative ground truth. Fig. 2(c) shows and the segmentation provided by the W4 algorithm (d). The later situations should be considered as errors.

Another ambiguous example is shown in Fig. 3. This suggests the use of multiple interpretations for the segmentation. To accomplish this the evaluation setup takes into account all possible merges of single regions belonging to the same group whenever multiple segmentation hypothesis may occur in the frame, i.e., when there is a small overlap among the group members.



$$\mathcal{C} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathcal{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathcal{C} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathcal{C} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 1. Different matching cases. Detection Failure (a); False alarm (b); Merge (c); Split (d).

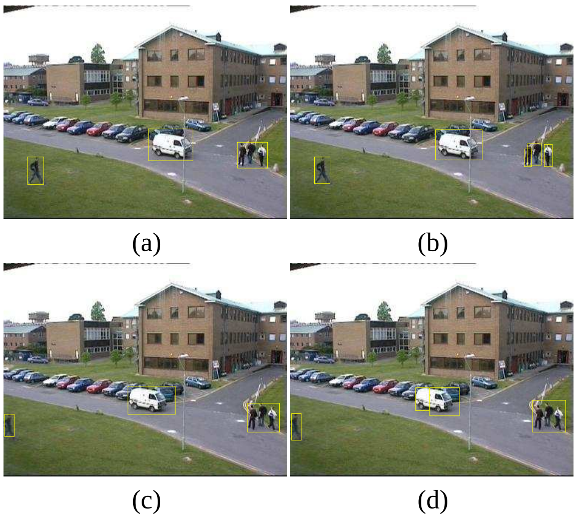


Fig. 2. Correct split example (first row), wrong split example (second row). Supervised segmentation (a), SGM segmentation (b); supervised segmentation (c), W4 segmentation (d).

The number of merges depends on the relative position of single regions. Fig. 4 shows different merged regions corresponding to a group of three objects (each one representing a person in the group) when the relative position is different. In Fig. 4 (a) it is not necessary to consider the merge of the first and the third regions since the second region is in the middle. However, if the positions of the regions change (see Fig. 4 (b)) the number of links may be different. It is reasonable to assume that each region can be merged with the all others. In the automatic evaluation process, it is enough for the user, to give single regions and the interpretations are generated automatically. Figs. 5,6 illustrate this situation. Fig. 5(a) shows the input frame, Fig. 5(b) shows the hand segmented image, and Fig. 5(c) illustrates the output of the SGM. Fig. 6 shows all possible merges of individual regions. All of them are considered as correct. Remain to know which segmentation should be selected to appraise the performance. In this paper we choose the best segmentation, which is the one that provides the highest number of correct detections. In the present example the segmentation illustrated in Fig. 6 (g) is selected. In this way we overcome the segmentation ambiguities that may appear without penalizing the algorithm.

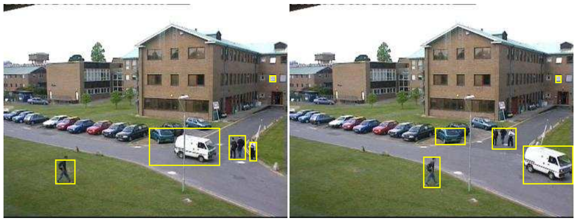


Fig. 3. Output of the SGM method at two time instants.

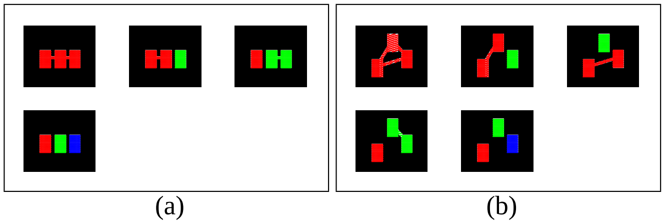


Fig. 4. Regions linking procedure. The same number of foreground regions may have different interpretations: three possible configurations (a), or four configurations (b). Each color represent a different region.

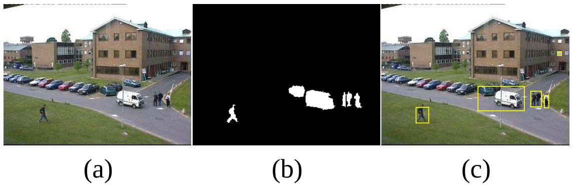
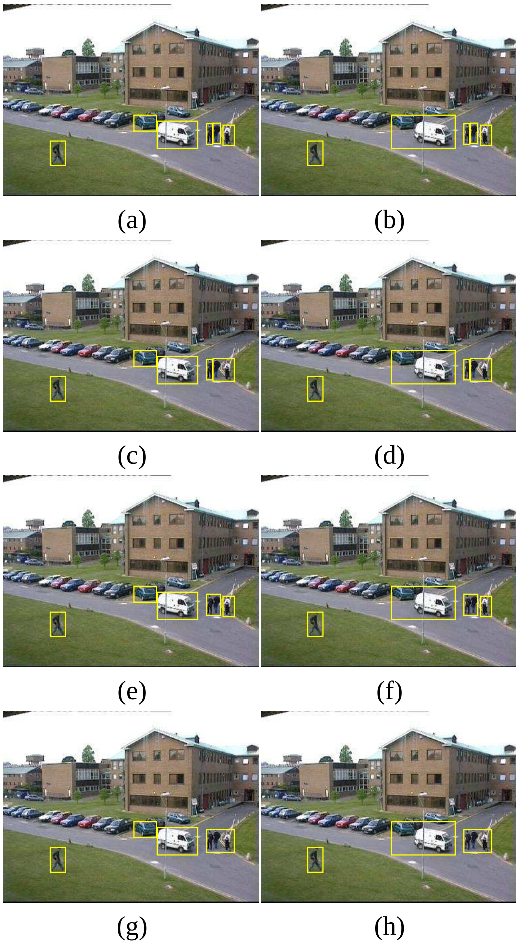


Fig. 5. Input frame (left image), segmented image by the user (middle image), output of SGM (right image).

5. TESTS ON PETS2001 DATASET

The segmentation algorithms described in this paper were evaluated using PETS2001 and PETS2004 dataset with resolution 384×288 pixels. The ground truth was generated by segmenting one image per second and correcting the automatic segmentation results using the graphical editor described before. The output of the algorithms was then compared with the ground



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 6. Multiple interpretations given by the application. The segmentation illustrated in (f) is selected for the current frame.

truth. The active regions with less than 25 pixels were eliminated.

The segmentation algorithms described herein depend on a set of parameters: thresholds and learning rate α. To find appropriate values for these parameters, we produced ROC curves which display the performance of each method as a function of the parameters. Each ROC is built by keeping all the parameters constant but one. This requires a considerable amount of tests, which were done using a training sequence of the PETS2001 data set. Once the parameters are set, we use these values to evaluate the algorithms in a test sequence of PETS2001.

ROC curves describe the evolution of the false alarms (FA) and detection failures (DF) as T varies. An ideal ROC should be close to the origin, i.e., with small area.

Fig. 7 shows the receiver operating curves (ROC) for the best value of α for different values of the threshold. These tests were performed on the training sequence. The BBS algorithm is sensitive to the threshold (see 7(a)). Small changes of T leads to large variations of false alarms and detection failures. The best value is $T = 0.2$. Fig. 7(b) shows the ROC of the SGM method, for $\alpha = 0.005$. This method is more robust than the BBS algorithm with respect to the threshold. A smooth variation of FA and DF is obtained for $-400 < T < -150$. We choose $T = -400$. Fig. 7(c) depicts the results of the MGM method. We notice that the algorithm strongly depends on the value of T, since for small variations of T there are significant changes of FA and DF. The best performance is obtained for $T > 0.9$.

Fig. 7(d) displays the results of the LOTS algorithm for a variation of the sensitivity from 10% to 110%. We use a small blending parameter. LOTS does not update the background image in every single frame to avoid a high computational cost. This algorithm only updates the background every N frames instead. We used an integration factor $\alpha = 6.1 \times 10^{-5}$ which corresponds to add 0.0625 of the current frame to the background in every 1024^{th} frame.

All methods have a large number of false alarms. In this sequence there is a static car which suddenly starts to move. Since the background is slowly updated, this event produces a ghost active object which is detected in a large number of frames.

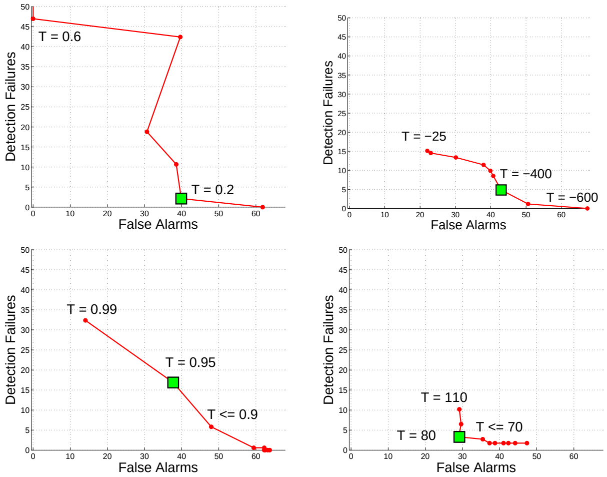


Fig. 7. Receiver Operation Characteristic for different values of α (training sequence). From left to right: BBS, $\alpha = 0.15$, SGM, $\alpha = 0.15$, MGM, $\alpha = 0.008$, LOTS, $\alpha = 6.1 \times 10^{-5}$.

Table 1 shows the results obtained on the test sequence using the parameters obtained from the ROC curves. In terms of false alarms the BBS method is the worst and the W4 is the best one. The main characteristics of the BBS method is that it tends to detect everything that moves in the scene. As a consequence it has a high percentage of correct detections but high false alarms rate.

The highest percentage of correct detections is achieved by LOTS followed by SGM. In the detection failures the W4 outperforms the others. W4 exhibits perhaps the best tradeoff between CD and FA. However, this method tends to split regions. This happens in situations in which the objects have a slow motion or when they stop, since the method is not able to remove the

	BBS	*W4*	*SGM*	*MGM*	*LOTS*
Correct Detections	83.5	76.7	87.9	74.8	**92.4**
Detection Failures	8.6	**4.8**	9.2	15.9	5.9
Splits	2.1	10.1	**0.2**	5.5	**0.2**
Merges	0	0.6	0	0	0.9
Split/Merges	6.2	6.9	3.0	3.4	**0.2**
False Alarms	21.7	**1.1**	11.1	10.7	7.9

Table 1. Performance of five object detection algorithms.

corresponding regions from the background model. This is the main drawback of the method. In term of merges, none of the algorithms studied here has a tendency to merge regions.

Comparing the results of false alarms between the Fig. 7 and the table 1, we notice that all the algorithms exhibit a larger percentage of false alarms in the training sequence (see Fig. 7) than in the test sequence (see table 1).

The choice of the method depends on the application. For instance the LOTS, SGM and W4 are well suited for tracking applications. Although the W4 method is sensible to splits, this is not a serious drawback in tracking since the system can always track one of the detected regions. However, if the application involves object recognition with global features (e.g., histograms) W4 is not suitable.

6. CONCLUSIONS AND FUTURE WORK

This paper proposes new metrics for the evaluation of object detection algorithms in surveillance applications. The proposed methodology is based on the comparison of the detector output with the ground truth segmentation of test sequences followed by a classification of the errors. The performance evaluation is made in terms of achieving the best tradeoff between correct detection and false alarms. Although we find LOTS and SGM the most suitable algorithms to perform region segmentation, the choice depends on the context of the application.

These tests should be further extended to consider other sequences. It would also interesting to enlarge the set of methods and to characterize the effect of each type of error on the performance of the overall system. The measurements proposed herein are important to characterize the performance of object detection algorithms.

Another important issue is that the proposed framework can also be used to evaluate tracking algorithms. To accomplish this, it is enough to record the trajectory of the detected regions using the ground truth.

Acknowledgement: We thank Prof. José Santos Victor for providing valuable information about the CAVIAR activities.

REFERENCES

Boult, T.E., R.J. Micheals, X. Gao and M. Eckmann (2001). Into the woods: Visual surveillance of non-cooperative camouflaged targets in complex outdoor settings. In: *Proceedings of the IEEE*. pp. 1382–1402.

Collins, R., A. Lipton and T. Kanade (1999). A system for video surveillance and monitoring. In: *Proc. American Nuclear Society (ANS) Eighth Int. Topical Meeting on Robotic and Remote Systems*. Pittsburgh, PA. pp. 25–29.

Cucchiara, R., C. Grana and A. Prati (2002). Detecting moving objects and their shadows: an evaluation with the PETS2002 dataset. In: *Proceedings of Third IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2002) in conj. with ECCV 2002*. Pittsburgh, PA. pp. 18–25.

Haritaoglu, I., D. Harwood and L. S. Davis (1998). W^4: Who? when? where? what? a real time system for detecting and tracking people. In: *IEEE International Conference on Automatic Face and Gesture Recognition*. pp. 222–227.

Haritaoglu, I., D. Harwood and L. S. Davis (2000). W^4: real-time surveillance of people and their activities. **22**(8), 809–830.

J. Nascimento, J. S. Marques (2004). New performance evaluation metrics for object detection algorithms. In: *PETS ECCV, 6th International Workshop on Performance Evaluation for Tracking and Surveillance*. Prague, Czech Republic. pp. 7–14.

McKenna, S. J., S. Jabri, Z. Duric, A. Rosenfeld and H. Wechsler (2000). Tracking groups of people. *Computer Vision and Image Understanding* **80**(1), 42–56.

Oberti, F., A. Teschioni and C. S. Regazzoni (1999). Roc curves for performance evaluation of video sequences processing systems for surveillance applications. In: *IEEE Int. Conf. on Image Processing*. Vol. 2. pp. 949–953.

Siebel, N. T. and S. J. Maybank (2001). Real-time tracking of pedestrians and vehicles. In: *Proc. of IEEE workshop on Performance Evaluation of tracking and surveillance*.

Stauffer, C., W. Eric and L. Grimson (2000). Learning patterns of activity using real-time tracking. **22**(8), 747–757.

Trees, H. Van (2001). *Detection, Estimation, and Modulation Theory*. John Wiley and Sons.

Wren, C. R., Ali Azarbayejani, T. Darrell and A. P. Pentland (1997). Pfinder: Real-time tracking of the human body. **19**(7), 780–785.