**Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Data Mining (CIDM 2007)**

Prediction of Abnormal Behaviors for Intelligent Video Surveillance Systems

Duarte Duque, Henrique Santos and Paulo Cortez
Information Systems Department
University of Minho
4800-058 Guimarães, Portugal
{duarteduque, hsantos, pcortez}@dsi.uminho.pt

Abstract–**The OBSERVER is a video surveillance system that detects and predicts abnormal behaviors aiming at the intelligent surveillance concept. The system acquires color images from a stationary video camera and applies state of the art algorithms to segment, track and classify moving objects. In this paper we present the behavior analysis module of the system. A novel method, called Dynamic Oriented Graph (DOG) is used to detect and predict abnormal behaviors, using real-time unsupervised learning. The DOG method characterizes observed actions by means of a structure of unidirectional connected nodes, each one defining a region in the hyperspace of attributes measured from the observed moving objects and having assigned a probability to generate an abnormal behavior. An experimental evaluation with synthetic data was held, where the DOG method outperforms the previously used N-ary Trees classifier.**

I. Introduction

With the continuous growth of the surveillance market, the price of the equipment tends to decrease. This fact, associated with the increasing need to secure people and goods, leads to a common scenario: a large number of video cameras are monitored by a single user.

With such an amount of data provided to the user, it is impracticable to visualize simultaneously the behavior of all the observed objects, in order to quickly and correctly detect danger situations. This reality provokes an overcharge of responsibility and loss of efficiency in the surveillance personnel, as stated in previous psychological studies carried out by [1] and [2] in Cambridge University and by military experiments [3]. Thus, automatic video surveillance systems are crucial for security professionals.

Aspiring the automatic detection and prediction of abnormal events a system, called OBSERVER [4,5,6], has been developed in the University of Minho. The system has endowed with state-of-the-art segmentation and tracking algorithms, and is able to classify tracked objects made of three classes: person, group of people and vehicles.

In this paper, we present the behavior analysis module of the OBSERVER. This novel classifier, called Dynamic Oriented Graph (DOG), is used to learn, detect and predict unusual and abnormal behaviors. The DOG classifier processes sequential data from tracked objects, signalizes unusual events and sends alarm warnings for possible abnormal behaviors.

This method constructs a structure to learn and maintain a set of observed patterns of activities, using real-time learning and without the requirement to perform any kind of training. This feature is an important advantage when compared with previous approaches in activity recognition from video surveillance data.

The paper is organized as follows. In Section 2 we briefly review some related work. Then, in Section 3, the main ideas behind the concept of the DOG classifier are explained. In Section 4 we present the proposed method to predict abnormal actions, and describe the implementation of the classifier. Experimental results are presented in Section 5 and, in Section 6, we draw some conclusions.

II. Related Work

Some attempts to automatically detect and predict abnormal behaviors were already performed. One of the pioneers was Ghallab [7], which has proposed a method to model temporal scenarios whose occurrences need to be recognized on-line. The author suggests a description of a generic scenario, of normal or abnormal behaviors, represented has a set of events and temporal constraints. The recognition algorithm implements propagation of temporal constraints with an incremental path consistency algorithm derived from the Mackword and Freuder work [8].

Several works on this field were also published under the ADVISOR project [9]. The aim of this project was to detect vandalism acts, crowding situations and street fights. To accomplish this task, it was necessary to build the entire three-dimensional model of the monitored area. In the ADVISOR system, danger behaviors were previously defined by security experts who described relevant events using a description language.

A more recent work carried out by Mecocci [10] introduces an architecture of an automatic real-time video surveillance system, capable of autonomously detecting anomalous behavioral events. The proposed system automatically adapts to different scenarios without human intervention, and applies self-learning techniques to automatically learn typical behavior of targets in each specific environment. Anomalous behaviors are detected whenever the observed trajectories deviate from the typical learned prototypes.

This work was supported by the FCT under the grant SFRH/BD/17259/2004

1-4244-0705-2/07/$20.00 ©2007 IEEE

Despite the considerable contribution, the system proposed by Mecocci does not accomplish the true requirements of a surveillance system, where typically the observed area comprises different levels of security. Another negative aspect of this approach is the use of simple spatial information from the center-of-mass position of the object, without considering other important features (e.g. color).

In the scope of the CAVIAR project, Nascimento et al. [11] proposed an algorithm to classify four simple human activities from a video sequence in a shopping center environment. Those activities, represented by the spatial position of the object over the time, consisted in the "*enter*" and "*exit*" from a store, "*pass in front*", and "*stop to watch*" the display window. This kind of approach involves a prior definition of the environment context, and the idea behind this method is based in the assumption that a behavior can be decomposed by a set of simple actions. In particular, five elementary actions were employed: "*move up*", "*move down*", "*stop*", "*move left*", and "*move right*". Thus, each elementary action was modeled by a Gaussian Model constructed from a training dataset of object displacements from elementary actions observed in successive frames. Then, after the training stage, the complex activities were classified through commutations between models of elementary actions. As an example, the authors describe the activity "*enter*" has a combination of the models "*move right*" followed by the "*move up*".

An attempt to perform unusual event detection and prediction of abnormal behaviors in public spaces was made in the OBSERVER project. The authors proposed the N-ary Trees behavior classifier [5], which was able to detect suspicious events without a prior definition of the behaviors of interest and, with a minimal description of the scene context. The N-ary Trees classifier generates the model of behaviors in a training stage. During the training, a considerable amount of data from previously observed tracks has to be processed by an Expectation-Maximization algorithm [12]. However, a considerable computational effort was required to construct the behavior model. Furthermore, once the models were generated it was impossible to make further adjustments to the learned patterns.

Despite the considerable achievements on the field accomplished in the recent years, there are still some challenges to overcome. The optimal behavior classifier should allow the detection of suspicious events with a minimal description of the scene context, perform the detection without the need of a training stage or dataset, and comprising the real-time constraints of a surveillance system. Furthermore, the classifier should be dynamic, i.e., a classifier that learns and adjusts itself to changes of object behaviors.

To achieve this goal, we developed a new behavior classifier, called Dynamic Oriented Graph, which is described and evaluated in this paper. This new approach to behavior classification and prediction will be tested against the recent N-ary Trees classifier.

III. Dynamic Oriented Graph Philosophy

Before explaining the DOG classifier, used to identify and predict object behaviors, we will begin by clarifying the idea behind the proposed classification system and its architecture.

One well accepted current in behavior analysis advocates that the behavior of an object can be described by a sequence of atomic actions that it performs in a certain environment [9,12]. This approach generally requires the knowledge from state history of the most representative and relevant attributes of the target object (e.g. hands, foots and head). Gathering the measures of these features can be difficult, and in many cases impossible, when dealing with real-life outdoor environments.

In real world video surveillance applications, objects can be far from the camera point of view and the existence of several objects moving in a scene can generate phenomenons like occlusions which can lead to errors or miss detections. Another difficulty in the implementation of this approach is that, despite the existence of well known methods to identify atomic actions, it is extremely challenging to recognize the start and stop points of such events. An extra obstacle is the demand for the description of the correct sequence of actions, which is mostly based on the user knowledge of the behaviors of interest.

Another proposal for behavior analysis, rather than fixate in a chain of simple actions, analyzes the entire time sequence in order to learn and cluster different patterns of activities [5,6,10]. This kind of perspective can be more robust, given that the behaviors are explained by a more general perspective rather than a restrictive sequence of actions. This more generalist approach requires however sophisticated methods for mining an huge amount of training data, in order to discover patterns that are common to certain types of behaviors. The resulting information is used to construct a model that explains such behaviors.

From the security point of view, we could expect three different kinds of behaviors: normal, unusual and abnormal. Typically, in an area under surveillance, the focus of attention falls into **unusual behaviors**, e.g. a person running in a hotel lobby; and violation of restricted areas (**abnormal behaviors**), e.g. a person crossing a railway. Hence, there is no need to define complex contexts of scenes or activities. The user only needs to define the masks of restricted regions from the scene.

Traditional techniques, that create the classifier model from a training dataset, can be used on-line but they are unable to do further adjustments to the learned patterns. Consequently, the quality and number of tracks of the dataset will influence the robustness and precision of the classifier in future measures. This can be critical in situations where the data changes rapidly making previously discovered patterns invalid.

The challenge is to define a dynamic classifier that learns and adjusts itself to changes of object behaviors with no need of a prior training phase. In order to support the user in the video surveillance task, an automated behavior classifier should also detect unusual behaviors but, most important, it has to predict the occurrence of abnormal actions, triggering an alarm that calls the user attention.

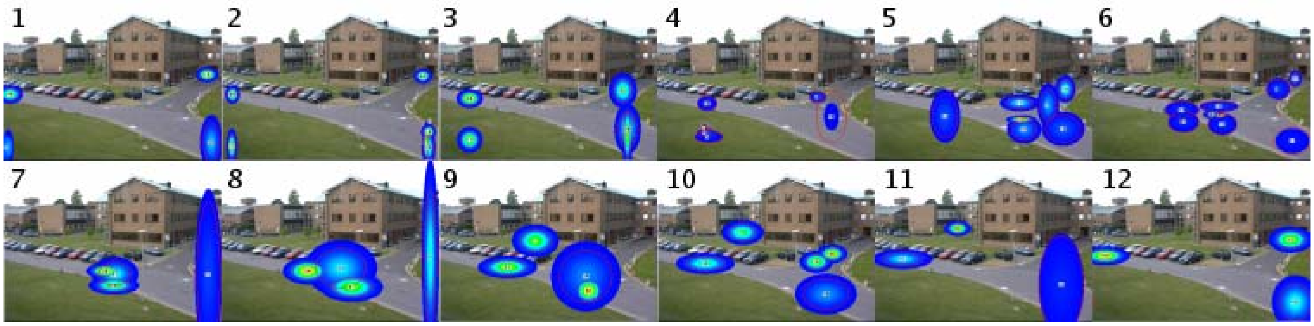

Fig. 1. Two-dimensional representation of the nodes of the first's twelve time segments from a set of tracks.

Another important feature is that the classifier must provide a way to allow human interpretation of learned patterns. Then, the data mining method should be able to model relevant patterns graphically.

After identifying the main features that an automated behavior classifier should fulfill, we will now present the data mining method. To do that, let's first focus our attention on the queries that the classifier has to answer. The detection of unusual behaviors and prediction of abnormal activities can be defined by the following questions: Is the trajectory of an object (with specific properties) known? If it is recognized, what is the probability to the object follow a path that will cause an abnormal event, based on its trajectory and properties history? Clearly, we can identify the need to discover new patterns in order to predict them, but we want to do that with a human-understandable description of the learned patterns.

The data supplied to the classifier comprises time and spatial information about the object trajectory, and other attributes regarding features like the area or perimeter of the object can also be considered. A technique to model behaviors can be achieved by using a structure of unidirectional connected nodes, each one defining a region in the hyperspace of attributes, for a fixed time length, measured from the observed moving objects and having assigned a probability to generate an abnormal behavior.

The resulting structure is an oriented graph composed by three types of nodes: entrance, path and departure nodes. The graph is ordered in time, with a set of nodes for each time segment. The entrance nodes define regions (in terms of normal distributions) in the hyperspaces of attributes at time zero, and departure nodes identify exit regions.

The main idea behind such classifier is that objects matching a pattern should belong to a node of the graph in the respective time segment. If the assignment to the model fails then an unusual event is detected.

With a dynamical structure, the model can adjust itself by merging nodes whose normal distributions enlarge, provoking their overlapping. The DOG classifier also reacts to the appearance of new tracks by including new nodes into the model. This dynamic update of the model is performed whenever is applied new input data to the classifier.

The ability to predict the probability of an object to follow a path that leads to a violation of a restricted area is achieved by computing the ratio between the amount of abnormal tracks that have passed by the node and the total number of tracks that get through the node.

Despite the simplicity of the concept, this approach to behavior classification comprises a simple mean to visualize the learned patterns. Furthermore, it enables the use of techniques to learn these patterns in an unsupervised approach, in real-time and, adjusts automatically to changes in the data.

IV. Generating the Dog Classifier

To put into practice our proposal, we have created a classifier that applies an oriented graph to structure common patterns of objects activities. The graph arranges the nodes by time, in a way that each layer of nodes defines the distributions of the objects properties during each time segment. In Figure 1 we can see an example of these distributions in twelve layers.

The partition of the data in segments of fixed time length can be seen as a mixture of a data compression technique with the discretization of the signals measured by the video analysis process. The compression of the data is supported by the computation of the mean value of each object attribute, over a time segment.

Based in the Nyquist-Shannon sampling theorem [13], the length of the time sequence should comply with the obligation that the sampling frequency has to be greater than twice the highest observed frequency. Transposing this theorem to our problem, we can state that the time length of the segments must be less than half the smallest fluctuations of attributes values (in the time domain), common to behaviors of interest.

Despite the theoretical foundations underlying the chosen of the optimal sampling frequency, in the context of our work, the length of the time segments were defined empirically. It should be clear that smaller time segments allow more precise behavior models. However, to the selection of the best possible value we recommend to balance this fact with the amount of memory available for tracing the models.

As we have stated before, the nodes in a layer intend to characterize the distributions of the objects properties in a given time segment. To perform this, each node defines a multivariable Gaussian distribution $N(\boldsymbol{\mu},\boldsymbol{\Sigma})$. These distributions are represented by a vector of mean values of the attributes ($\boldsymbol{\mu}$) and a covariance matrix ($\boldsymbol{\Sigma}$). In order to simplify the computation, we assume that the variances in the covariance matrix are not correlated.

Another important feature of the nodes is the **Probability of Abnormal Behaviors** (*Pab*). In other words, when an object transits from a node to another, following a path according to the spatial and time relations of the model, it is possible to predict the probability that the object will violate a restricted area.

A. Defining the Parameters of the DOG Classifier

The first step required by the DOG classifier is the selection of restricted areas. This is made for each class of objects (person, group and vehicle). In the Figure 2, we show an example of restricted areas masks for vehicles and people. Different masks can share common regions of the scene.

In addition, we carry out with the choice of the appropriate length of the **Time Segments** (*Ts*) and the default values for standard deviation ($\boldsymbol{\sigma}_{def}$) to assign in new nodes. Note that the default values for the standard deviations can be different, reflecting the nature of the attributes.

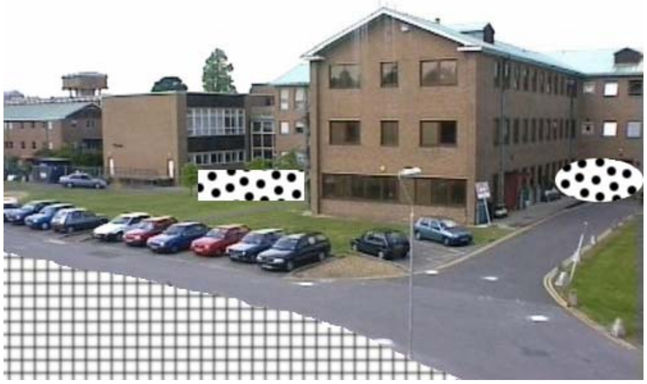

Fig. 2. Example of restricted areas for vehicles (lattice) and people (dots).

B. On-line Modeling of Behaviors by the DOG Classifier

When initiated at the first time, the model of the DOG classifier has no information about any kind of behaviors. The model is an empty instance. Then, each provided data sequence, describing the properties of the spatial motion, size, and colors of an object, will be therefore used to add knowledge to the model about the observed behavior.

The sequential data from an object enters in the classifier through a buffer (***b***) with a length equal to *Ts*. When the buffer reaches to its maximum capacity, the mean values of the attributes are computed, and the results are saved in the input vector (***v***). This operation is described by:

$$\begin{array}{l} foreach \quad n \in N \quad do \\ \qquad v(n) = \dfrac{\sum_{i=1}^{Ts} b(i,n)}{Ts} \\ end \end{array} \tag{1}$$

where *N* defines the number of attributes extracted from the object.

The next step is to verify if the input vector can be assigned to one of the *D* distributions from the first layer. The condition for validating the assignment must satisfy the following criteria:

$$\begin{cases} assigned \quad ,if \quad |v(n) - \mu_d(n)| \le \sigma_d(n) \quad ,\forall n \in N \wedge \exists d \in D \\ unassigned \quad ,otherwise \end{cases} \tag{2}$$

If the assignment is well succeeded the mean vector and the standard deviation of the selected distribution are updated according to the rules:

$$\begin{array}{l} foreach \quad n \in N \quad do \\ \qquad \mu_d(n) = (1-\alpha).\mu_d(n) + \alpha.v(n) \\ \qquad \sigma_d(n) = (1-\alpha).\sigma_d(n) + \alpha.|v(n) - \mu_d(n)| \\ end \end{array} \tag{3}$$

If, for other hand, the input vector can’t be assigned to any of the available nodes in the considered time segment, then a new node is created. In this case, the Gaussian distribution will be generated by a covariance matrix based on the default values for the standard deviation ($\boldsymbol{\sigma}_{def}$), and a mean vector receiving the values of the input vector ($\boldsymbol{\mu}_d$=***v***). Next, the new node must be linked to the parent nodes.

After updating or creating new nodes, the model should be scanned in order to detect the merging of nodes. A node is merged to other of the same layer if their Gaussian distributions overlap each other. When merging the nodes, the resulting distribution must be computed by:

$$\begin{array}{l} foreach \quad n \in N \quad do \\ \sigma(n) = \dfrac{MAX(\mu_1(n)+\sigma_1(n), \mu_2(n)+\sigma_2(n)) - MIN(\mu_1(n)-\sigma_1(n), \mu_2(n)-\sigma_2(n))}{2} \\ \mu(n) = \dfrac{\mu_1(n).Tracks_1 + \mu_2(n).Tracks_2}{Tracks_1 + Tracks_2} \\ end \end{array} \tag{4}$$

where $Tracks_1$ and $Tracks_2$ defines respectively the total number of tracks passing over the “*node 1*” and “*node 2*”.

The final step is concerned to updating the Probability of Abnormal Behaviors (*Pab*). This is made when an object enters into a restricted area or when it leaves the scene. In the case of a restricted area violation, the set of nodes crossed by the object will increment the track counter (*Tracks*) and the abnormal track counter (*AbTracks*). The *Pab* is therefore computed by the following ratio:

$$P_{ab} = \frac{AbTracks}{Tracks} \tag{5}$$

If an object leaves the scene without overlapping a restricted area, then only the track counter of their path nodes will be incremented. These counters (*Tracks* and *AbTracks*) are normalized by a maximum allowed value, in order to prevent the counters overflow. A three-dimensional representation of a DOG model can be seen in Figure 3.

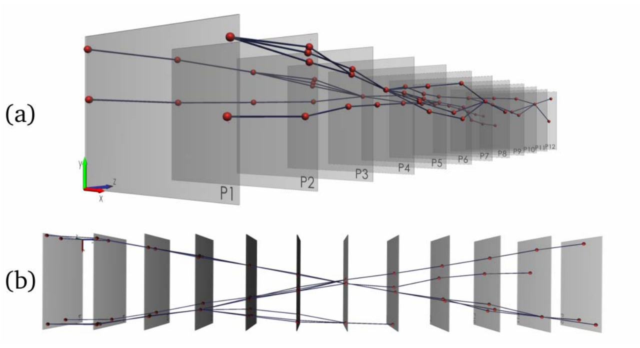

Fig. 3. Three-dimensional representations of the DOG model presented in figure 1. (a) Perspective view; (b) Top view.

C. *Classifying Unusual and Abnormal Behaviors*

Each tracked object has a pointer to the model, defining in which node he is assigned. The transition for another node in the next layer must obey to the restrictions imposed by the connections of the current node. If during the transition none of the allowed nodes is assigned, then a warning of **unusual event** is signed.

Abnormal behaviors can be predicted by defining an **Alarm Threshold** (A_t) for the model. Then, when an object arrives to a node whose P_{ab} is higher than that threshold, an **alarm event** is triggered, pointing to the possibility of a restricted area violation.

VI. Experimental Results

A dataset of one thousand synthetic tracks with different lengths was created based on a scene with two restricted areas, as illustrated in Figure 4. The dataset is composed by a label identifying each observed object and the center-of-mass coordinates along the time. In these tests, no color or morphological information was used, as we consider that these features will be relevant only for finding patterns in the presence of real data.

In order to make the dataset more challenging, some white noise (with 3 pixels of standard deviation) was added to the data. In addition, the speed parameter was induced by a Gaussian noise with a variance of 6 pixels.

The generated tracks represent sixteen different paths of a same kind of objects (people) with two hundred of the tracks violating a restricted area. Since the N-ary Trees and DOG classifiers made use of a model for each type of object (people, groups and vehicles), we only need to perform the tests for one type of object, i.e. people, in order to evaluate the performance of both classifiers.

To evaluate the accuracy of the proposed classifier against the N-ary Trees, a 10-fold cross validation scheme [14] was adopted. In the evaluation of the DOG classifier, at each step of the cross validation, 9/10 of the dataset was used for learning the model dynamically and the remaining 1/10 for testing the model, with the dynamic update disabled. For the N-ary Trees, the process was similar, performing a training phase with 9/10 of the data and applying the remaining data to test the model. In order to increase the confidence level of the evaluation results, we performed thirty different runs of the 10-fold for each classifier.

The evaluation tests for the DOG and the N-ary Trees classifiers were run on the same conditions in a standard computer, based on a 32 bits 3 GHz processor equipped with a GNU/linux operative system. To test each classifier it was necessary to execute the entire thirty runs of the 10-fold cross validation, which gives a sum of three millions processed tracks, i.e., one thousand tracks by thirty runs by one hundred Alarm Thresholds (A_t).

For evaluation purposes the **Receiver Operating Characteristic** (ROC) and the **Anticipation Rate** curves were computed. These values for the ROC curve were calculated by constructing the Confusion Matrix for each value of the A_t (ranging from 0% to 100%, in steps of 1%). The Anticipation Rate is given by the ratio between the anticipation distance (the distance from the point of prediction of abnormal behavior to the point of violation of a restricted area) and the total length from the beginning of the track until the overlap of the restricted area. The Anticipation Rate curve is computed for different A_t values, similarly to the ROC computation.

The simulation for the DOG classifier required one minute and twenty nine seconds to execute, while the execution of the N-ary Trees demanded a much higher computation, approximately one hundred and ninety hours. This enormous disparity of time required for testing the classifiers can be explained by the considerable computational cost required by the N-ary Trees classifier in the training stage.

In the Figure 5, we show the reaction of the DOG and the N-ary Trees classifiers for the range of alarm threshold values (A_t). As we can see, the True Positive Rate (TP Rate) drops when the A_t value is increased. In addition, the Anticipation Rate decreases for higher values of A_t. This behavior is common to both classifiers. Another relevant metric to quantify the performance of the classifiers is the Area-Under-the-Curve (AUC). The AUC measures the classifier capability, and allows a comparative evaluation with other classifiers processing the same dataset. With the dataset used in these experiments, we achieve an AUC of 86% for the DOG classifier and an AUC of 79% for the N-ary Trees approach.

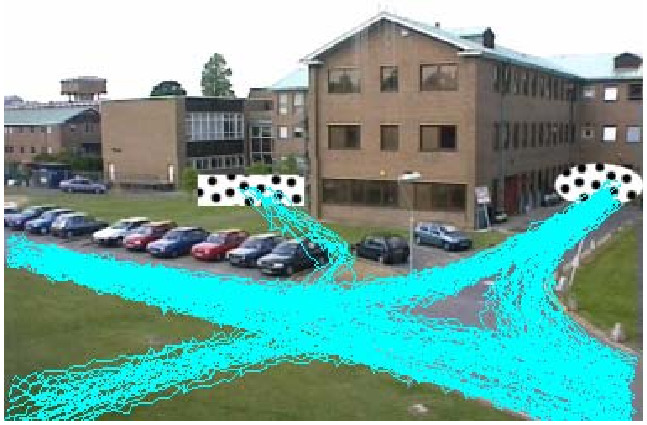

Fig. 4. Some of the synthetic generated tracks of people dataset.

With the information provided in the Figure 5, it is straightforward to the user make a decision for the ideal value for *At*. The combined information from the ROC and the Anticipation Rate curves, allows a decision of the optimal value, based on the best compromise of False Positive Rate, True Positive Rate and the required level of prediction.

VII. Conclusions

In this work, a new approach to automatically detect and predict abnormal behaviors was presented. The behavior of each type of object (person, group or vehicle) is modeled by a single DOG. Under this scheme, it is possible to assign distinct behaviors to different kind of objects performing a similar path. For example, crossing a garden can be a normal behavior for humans, but an unusual one for vehicles.

A relevant aspect of the proposed classifier is that it has revealed to be appropriate in surveillance tasks aiming to prevent the violation of restricted areas. Another important feature is based in the fact that the detection of unusual events, of relevance to call the attention of the system user for possible illicit behaviors, has the advantage to be absolutely context free. The DOG classifier demonstrates to be extremely fast, learning, classifying and predicting activities on-line and in a dynamical form. Therefore, the classifier shows to be suitable for use in products for the video surveillance market (intelligent cameras and digital video recorders).

Despite the encouraging results achieved by the DOG classifier, it is imperative to make note that it was designed to handle objects individually, i.e., the classifier does not detect danger situations from combinations of multiple objects and their interactions. However, the classifier is still able to detect, simultaneously and in real-time, the behavior of a very large number of objects.

In future work, we will try to test the DOG classifier with real-data in an extensive range of environments. A complementary study, to define which attributes are of relevance for behavior detection should also be carried out.

References

[1] A. H. Tickner, E. C. Poulton, A. K. Copeman and D. C. V. Simmonds "Monitoring 16 television screens showing little movement," *Ergonomics*, Vol. 15, pp. 279-291, 1972.

[2] A. H. Tickner and E. C. Poulton, "Monitoring up to 16 synthetic television pictures showing a great deal of movement," *Ergonomics*, vol. 16, n. 4, pp. 381-401, 1973.

[3] T. Ainsworth, "CCTV – Performance testing," *Security OZ Magazine*, Issue 9, pp. 50-59, 2004.

[4] D. Duque, H. Santos and P. Cortez, "Moving object detection unaffected by cast shadows, highlights and ghosts," *Proceedings of the IEEE International Conference on Image Processing*, Genoa, Italy, pp. 413-416, 2005.

[5] D. Duque, H. Santos and P. Cortez, "N-ary Trees Classifier," *Proceedings of the 3rd International Conference on Informatics in Control, Automation and Robotics*, Setúbal, Portugal, pp. 2006.

[6] D. Duque, H. Santos and P. Cortez, "The OBSERVER: An Intelligent and Automated Video Surveillance System," *Proceedings of the International Conference on Image Analysis and Recognition*, Póvoa de Varzim, Portugal, pp. 989-909, 2006.

[7] M. Ghallab, "On Chronicles: Representation, On-line Recognition and Learning," *5th International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, pp. 597-606, 1996.

[8] A. K. Mackworth and E. C. Freuder, "The Complexity of Some Polynomial Network Consistency Algorithms for Satisfaction Problems," *Artificial Intelligence*, 25, pp. 65-74, 1985.

[9] M. Naylor. (2006) ADVISOR project website. [Online]. Available: http://www-sop.inria.fr/orion/ADVISOR/

[10] A. Mecocci, M. Pannozzo and A. Fumarola, "Automatic Detection of Anomalous Behavioural Events for Advanced Real-Time Video Surveilance," *International Symposium on Computational Intelligence for Measurement Systems and Applications*, Lugano, pp. 187-192, 2003.

[11] J. C. Nascimento, M. A. T. Figueiredo and J. S. Marques, "Segmentation and classification of human activities," *International Workshop on Human Activity Recognition and Modeling*, Oxford, UK, 2005.

[12] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, Series B, vol. 39, pp. 1-38, 1977.

[13] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the Institute of Radio Engineers*, vol. 37, no.1, pp. 10-21, 1949.

[14] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, San Francisco, pp. 1137-1143, 1995.

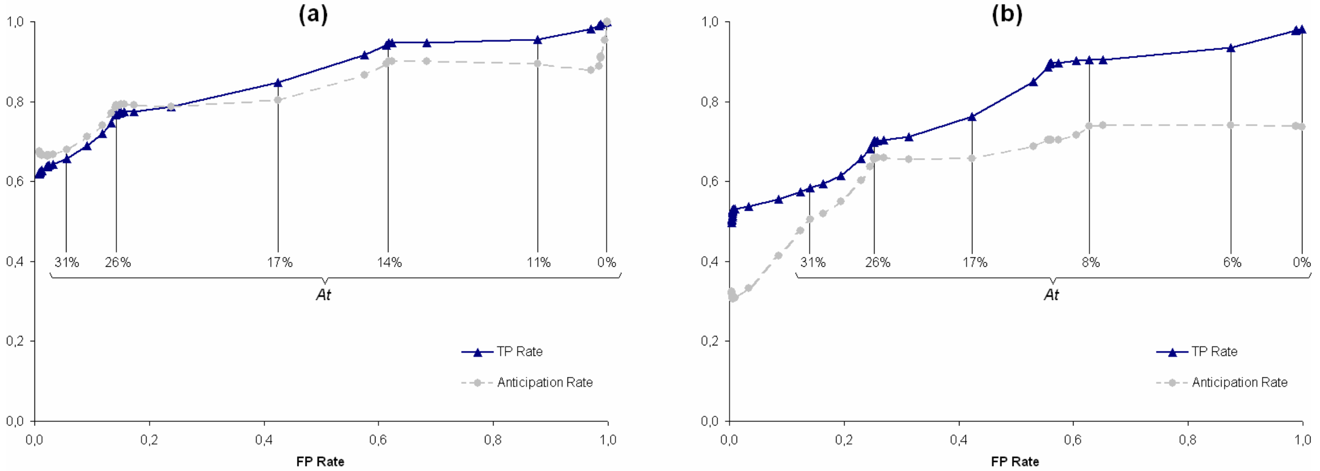

Fig. 5 Receiver Operating Characteristic curves overlapped by the Anticipation Rate curves for: (a) the DOG classifier; (b) the N-ary Trees classifier.