Selection of Temporal Features for Event Detection in Smart Security

Niki Martinel[1(✉)], Danilo Avola[1], Claudio Piciarelli[1], Christian Micheloni[1], Marco Vernier[1], Luigi Cinque[2], and Gian Luca Foresti[1]

[1] Department of Mathematics and Computer Science, University of Udine,
Via delle Scienze 206, 33100 Udine, Italy
niki.martinel@uniud.it

[2] Department of Computer Science, Sapienza University of Rome,
Via Salaria 113, 00198 Rome, Italy

Abstract. Scene understanding in smart surveillance and security is one of the major fields of investigation in computer vision research and industry. The ability of a system to automatically analyze and learn the events that occur within a scene (e.g., a running person, a parking car) is conditioned by several complex aspects such as feature extraction, tracking and recognition. One of the most important aspects in the event learning process is the detection of the time interval in which an event occurs (i.e., when it starts and ends). The present paper is focused on the learning of temporal correlated events. In particular, a formalized description of the features associated with each event and the linked strategy to define the event time-line are provided. The paper also reports preliminary tests carried out on videos related to a reference outdoor environment which validate the proposed strategy.

1 Introduction

Nowadays, intelligent video surveillance systems (e.g., [1,2]) are becoming increasingly important due to their strategic role in monitoring sensitive targets (e.g., [3,4]), as well as in supporting the safety and security of the people, environments [5] and objects. The main task of these systems is to automatically interpret the behavior of the agents (e.g., persons, vehicles) that act within the scenario in order to detect suspicious events [6]. In fact, a first basic classification of the events can be defined by considering usual events and unusual events (e.g., anomalies or novelties) [7,8]. The first class takes into account all those events which are consistent with the monitored environment or, in other words, which are statistically more frequent. An unusual event belongs to the second class. In most situations, the primary objective of a smart security system is to provoke a reaction (e.g., alarm, feedback, action) when unusual events are perceived since they can potentially cause security violations [9]. It should be noted that an event considered to be usual in one environment could be an unusual event in other sites. Moreover, within the same event may be behaviors attributable to one or the other class. The introduced issues implicitly highlight two crucial aspects of the event detection

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part II, LNCS 9280, pp. 609–619, 2015.
DOI: 10.1007/978-3-319-23234-8_56

process. The first one regards the set of key features chosen to represent an event and its temporal evolution. The last one concerns the recognition of the time interval in which the event occurs. Note that, as previously mentioned, the same event (e.g., a moving car) can present, at different time intervals, features attributable to usual or unusual events thus complicating the already complex task of the event classification. From an abstract point of view, an algorithm for event detection and classification [10–13] consists of two main steps:

- The selection of the key features (i.e., feature selection technique [14]) through which to understand the semantic of the interesting events;
- The design of the classifier (i.e., machine learning technique) adopted to recognize the events. Note that, a classifier can be binary (distinguishing usual events from unusual events), or multiclass (characterizing type and meaning of each event within a set of specified classes).

The present paper has two main aims. The first one is to introduce a formalization related to the key features associated to any event. The last one is to exploit this formalization to introduce a time interval based strategy through which automatically recognize when an event starts and ends. The proposed approach was stressed by using a reference scenario in which different agents interacted forming a wide range of events. The obtained qualitative and quantitative results shown the effectiveness and the accuracy of the method.

2 Related Work

Due to the heterogeneity and vastness of the literature on the event representation and key feature extraction, the section will focus only on these works which are more directly related to the proposed approach. A first interesting work is presented in [15], where the authors address the problem of trajectory analysis for anomaly detection using SVMs. In their approach the key features are represented by a set of fixed-dimension vectors which correspond to the paths traveled by agents within the monitored scenario. The trajectories having similar features are clustered together forming the class of the usual events, while the other trajectories (i.e., outliers) will form the class of the unusual events. In [16], the role of the temporal feature for detecting unusual events is highlighted. The representation of the time is achieved by a Gaussian Mixture Model (GMM) which describes the temporal evolution of each agent involved within the scenario. Another challenging work is proposed in [17], where the authors describe a system for image understanding of complex scenarios. In particu-lar, the authors primarily adopt two key features (i.e., color and texture) to determinate the individual components of the images, subsequently they use another key feature (i.e., shape) to identify the important characteristic traits of the image and hence help in better descriptive analysis of the scene. The authors of the work shown in [18] adopt a classifier based on a set of DBNs to develop a situation assessment framework. In particular, they present a method for an automatic definition of the parameters that can be easily used by a human operator when designing a new net-work. The key features adopted within the framework come from a supervised data-base (VIRAT [19]) able to provide a semantic interpretation (labels) of the observed scenes, while the authors introduce an interesting use of the temporal feature. The temporal key feature is used to correlate different actions related to different agents with the aim to derive a more complex interpretation of the events. This work has the great advantage of being particularly adaptable to different application contexts. Other three interesting works are proposed in [20–22]. The first one presents an algorithm for learning the event categorizations in a given scene. In particular, the authors propose a Stochastic Context-Free Grammars (SCFG) for event detection in challenging outdoor video sequences. A similar work is introduced by the authors of the second one, which show an algorithm for recognizing usual and unusual events in complex video sequences. A technique of the same type is adopted by the authors of the last work, which propose both standard visual key features and domain-specific in-formation to semantically interpret different actions from video sequences. In each work the temporal feature is used to correlate spatial events and/or identify the sequence of complex actions.

3 Preliminaries

Let consider a monitored environment $\mathcal{S}$ in which n_A agents (i.e., objects), interact with it or among them. The k-th agent of class c, which is observed in the scenario at time instant t, is denoted as $\mathbf{a}_{(c,k)}(t)$ and it is composed of a set of n different features such that:

$$\mathbf{a}_{c,k}(t) = \{\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \ldots, \mathbf{x}_k^{(j)}, \ldots, \mathbf{x}_k^{(n)}\} \tag{1}$$

where $\mathbf{x}_k^{(j)} \in \mathbb{R}^d$ indicates the j-th feature type (e.g., color histogram, histogram of oriented gradients, etc.) extracted at time instant t (omitted to ease the notation). Notice that, since different features can be adopted to represent an agent, then it is not a necessary condition that the i-th and j-th features span the same feature space, i.e.$|\mathbf{x}_k^{(i)}| = |\mathbf{x}_k^{(j)}|$. In the following we assume all the features to be normalized in the interval $[0, 1]$.

3.1 Events

An event is generated by the action performed by a single agent. Therefore, it considers no interactions between agents. More formally, let $\mathbf{e}_{(c,k)} \in \boldsymbol{\varepsilon}_c = \{1, \ldots, N\}$ denote the type of an event of interest generated by the action of an agent $\mathbf{a}_{c,k}$ (e.g., a moving person, a parked car, etc.), and $\boldsymbol{\varepsilon}_c$ is the set of all N possible events associated with the c-th class agent. The set $\boldsymbol{\varepsilon}_c$ is defined off-line by the system operator and stored in an event database. Also, let $T^s = [t_0^s, t_{\text{end}}^s]$ denote the temporal interval during which an agent $\mathbf{a}_{c,k}$ performs an action of interest that fires an event. Given the set of all available features collected during the temporal interval T^s, denoted as:

$$\mathcal{A}_{c,k} = \{\mathbf{a}_{c,k}(t)\}_{t=t_0^s}^{t_{\text{end}}^s} \tag{2}$$

it is generally the case that only a subset of them is relevant to detect the event of interest. Such a subset is denoted as:

$$\tilde{\mathcal{A}}_{c,k} = \{\tilde{\mathbf{a}}_{c,k}(t)\}_{t=t_0^s}^{t_{\text{end}}^s} \subset \mathcal{A}_{c,k} \tag{3}$$

where $\tilde{\mathbf{a}}_{c,k}(t) \subset \mathbf{a}_{c,k}(t)$. Therefore a simple event generated by the k-th agent of class c, can be defined as:

$$\mathbf{e}_{c,k} = f_c(\tilde{\mathcal{A}}_{c,k}) \tag{4}$$

where $f_c(\cdot)$ is an agent class dependent function of the relevant features in $\mathcal{A}_{c,k}$ and which output in ε_c denotes the event type for the c-th class. For instance, the function $f_c(\cdot)$ can be the output of a classification algorithm (e.g., a Support Vector Machine, a Random Forest of Decision Trees, etc.).

4 Temporal Event Learning

One of the most important tasks that are required to properly learn and recognize an event of interest is to determine the time interval $T^s = [t_0^s, t_{end}^s]$ in which the event occur. To determine such a time interval we formulate the following hypothesis. Let t_0^s be the time instant in which the agent $\mathbf{a}_{(c,k)}$ enters the environment or the time instant following the last time instant of another event produced by the same agent. To establish the time instant t_{end}^s at which the event ends, we suppose that the given relevant features $\mathbf{a}_{(c,k)}$ do not vary over a given threshold. More formally, t_{end}^s is computed by taking the time instant t_i^s as follows:

$$\begin{cases} t_{\text{end}}^s = t_i^s & \text{if } h_c = 1 \\ t_i^s = t_{i+1}^s & \text{otherwise} \end{cases} \tag{5}$$

where

$$h_c = \left(\tilde{\mathbf{a}}_{c,k}(t_i^s), \tilde{\mathbf{a}}_{c,k}(t_{i-1}^s)\right), j = 1, \ldots, N \tag{6}$$

is a function that compares the value of the given relevant features to determine when the variation between two consecutive time instants t_i^s and t_{i-1}^s is outside the allowed range of $K_c^j = [\epsilon_{c,\min}^j, \epsilon_{c,\max}^j]$ which describes the behavior of the agent $\mathbf{a}_{(c,k)}$, hence the main characteristics of the event. For example, let us consider the detected agent $\mathbf{a}_{\text{person},1}(t)$ being a person in the environment. Let consider the following events belonging to the scenario:

- Event 1: A walking person (moving at approximately constant speed);
- Event 2: A running person;
- Event 3: A static person.

In such a case we can defined the function h_{person} as:

$$\begin{aligned} h_{\text{person}} &= \left(\tilde{\mathbf{a}}_{\text{person},1}(t_i^s), \tilde{\mathbf{a}}_{\text{person},1}(t_{i-1}^s), \epsilon_{\text{person}}^j\right) \\ &= \tilde{\mathbf{a}}_{\text{person},1}(t_i^s) - \tilde{\mathbf{a}}_{\text{person},1}(t_{i-1}^s) \in K_{\text{person}}^j \end{aligned}$$

where t_i^s is the sampling instant. In such a case the sampling interval $t_i^s - t_{i-1}^s$ is the 1sec. The value of the thresholds intervals for K_c^j , for $j = 1, \ldots, 3$ (in this example) should be selected according to the definitions of simple events. Here, the most significant feature characterizing the events is the speed of the person expressed as the distance traveled between consecutive frames measured in pixel or meters (if camera calibration is adopted). So, the set $\mathbf{a}_{\text{person},1}$ contains the agent position features only. The intervals K_c^j accounts for the minimum and maximum distance traveled between two consecutive frames for the j-th event of the agent of class c. For the given example, assuming that a walking person is normally moving at a speed lower than 2m/sec (i.e., covering a distance lower than 2m between consecutive samples), and that a running person is normally moving at a speed higher than 2m/sec, the following intervals can be defined by the operator for the event related to the agent $\mathbf{a}_{\text{person},1}$:

- Event 1: A walking person (moving at approximately constant speed): $\epsilon_{\text{person,min}}^1 = 0.25$ and $\epsilon_{\text{person,max}}^1 = 2$;
- Event 2: A running person: $\epsilon_{\text{person,min}}^2 = 2$ and $\epsilon_{\text{person,max}}^2 = 10$;
- Event 3: A static person: $\epsilon_{\text{person,min}}^3 = 0$ and $\epsilon_{\text{person,max}}^3 = 0.25$;

4.1 Temporal Feature Extraction

Typical features describing an event are the class of the agent, the size of the blob representing the agent, its position projected on a 2D top view map, the agent speed and the color histogram (see Fig. 1). In addition, to capture the shape of the agent [23] PHOG features are used. Such as features are extracted every time instant t and pooled in the time interval The PHOG feature [24] captures the local shape and the spatial layout of the shape in a given image exploiting the pyramidal framework proposed in [25]. As shown in Fig. 2, in a spatial pyramid framework, the given image is divided into a sequence of spatial grid cells. The PHOG feature vector is computed as a concatenation of all the HOG vectors computed for all the cells at each level of the spatial pyramid representation. Fig. 2 illustrates this principle showing the PHOG features computed for different values of spatial pyramid levels L.

4.2 Neural Tree Learning

A Neural Tree is a hierarchical classifier with a tree structure. Each node of the tree is an Artificial Neural Network (ANN) and each leaf corresponds to a class label (in this case, to an event). In a NT, the classification of a given

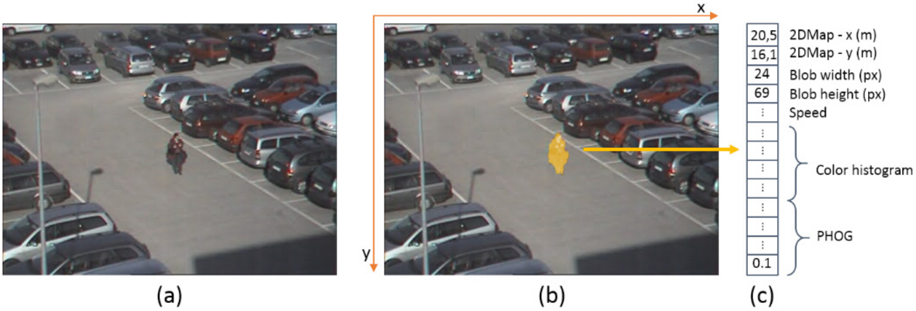

Fig. 1. An example of the considered features extracted from a given frame. (a) Shows the input frame; (b) The processed frame by a change detection algorithm localizing the agent; (c) The extracted feature vector.

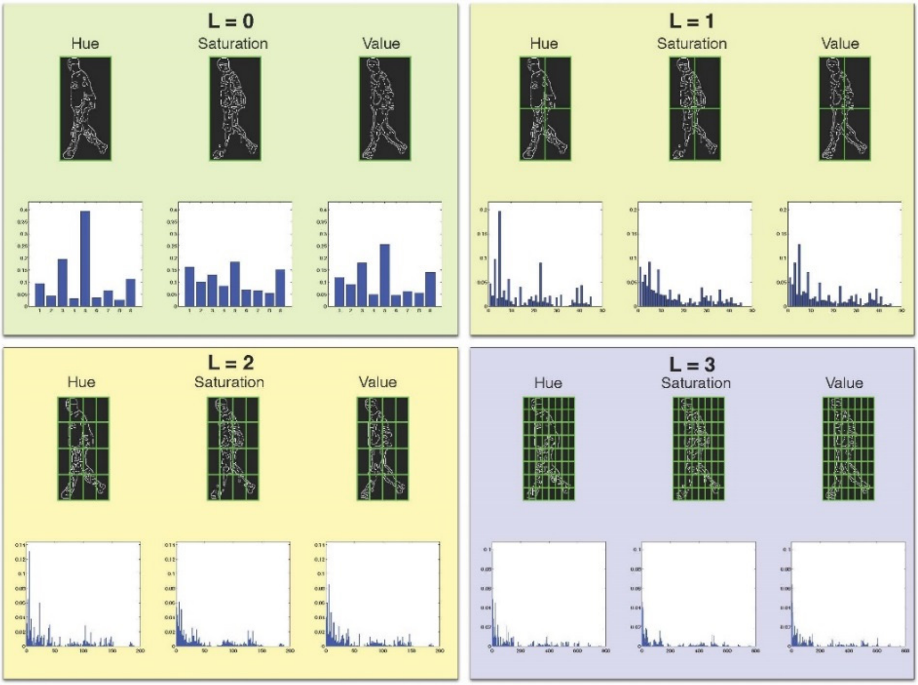

Fig. 2. Effects of the number of levels (L) in the PHOG feature extraction process. For each of the four blocks, the top row shows the grid cells (in green) at which the HOG features are extracted. Bottom rows show the final PHOG features computed concatenating the HOG features extracted at each level of the pyramid.

pattern is conducted by presenting the pattern to the ANN in the root node. Then, it follows the path determined by the ANN output. The process continues until a leaf node is reached. Due to the its classification capabilities, in this work the AHNT proposed in [26] has been used to obtain the classification function $f_c(\cdot)$. In an AHNT, each node can be a first-order or a high order perceptron (HOP) according to the complexity of the local training set. First order perceptrons split the training set by hyperplanes, while n-order perceptrons use n-dimensional surfaces. An adaptive procedure decides the best order of the HOP to be applied at a given node of the tree. The AHNT is grown automatically during the learning phase: its hybrid structure guarantees a reduction of the number of internal nodes with respect to classical neural trees and reaches a greater generalization capability. Moreover, it overcomes the classical problems of feed-forward neural networks (e.g., multilayer perceptrons) since both types of perceptrons does not require any a-priori information about the number of neurons, hidden layers, or neuron connections.

5 Experimental Results

In order to evaluate the performances of the proposed event learning and detection system in a real-world outdoor scenario, we considered the premises of the Department of Mathematics and Computer Science at the University of Udine and their surroundings, as shown in Fig. 3.

The environment consists of two main buildings, three parking lots, several roads and road joints, three main pedestrian-only paths and green areas. These areas have been manually labeled on the map (see again Fig. 3) and they compose the prior static knowledge available on the monitored environment. The area is monitored by six PTZ IP color cameras manufactured by Axis (outdoor model Q6032-E). The cameras have been oriented using the algorithm proposed in [27] to maximize their visual coverage avoiding overlaps, in order to acquire data from the largest area possible. The list of possible agents is defined as $A = \{person, car, bus\}$ and for each agent the list of features is defined as:

- $\mathbf{a} = \{position, speed, width, height, colorhistogram, PHOG\}$.

In order to learn the simple events with the AHNT [26] we considered the video sequences acquired over a 6-hours time range; the sequences have a resolution of 382×288 pixels. The experiments have been conducted to recognize three types of simple events, namely:

- Person walking on the pedestrian paths;
- Car entering/leaving the parking lot;
- Bus stopping/starting at the bus stop.

As previously mentioned, simple events are learnt using a neural tree architecture, using the most appropriate features among the available ones. The training set consisted in 451 sequences of walking person, 92 of car entering/exiting the

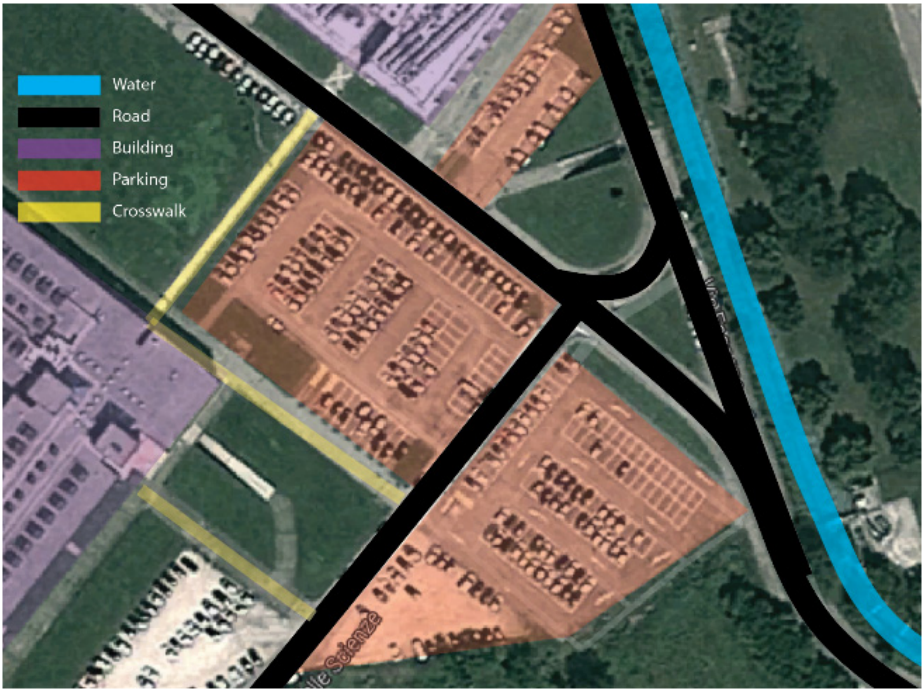

Fig. 3. The outdoor environment used as a test bed for simple/complex event learning and detection evaluation.

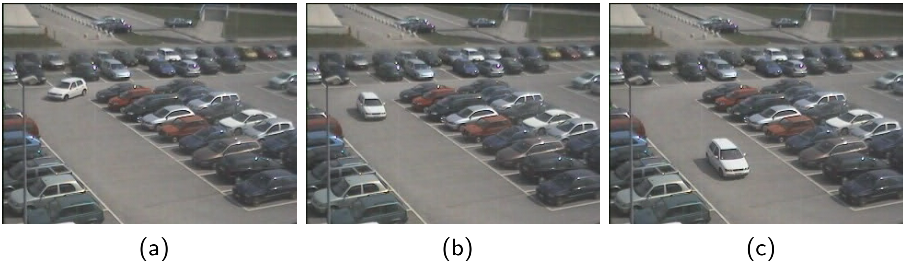

Fig. 4. Three frames from one of the parking lot sequences.

parking lot, and 12 of the stopping/starting bus. Fig. 4 shows few frames from a sequence in the parking lot.

The trained Neural Tree has 25 perceptron nodes and a maximum depth of 4. It has been used to classify respectively 50, 20 and 10 sequences for each of the three event classes (the test sequences are different from the training ones). The final classification result is show in Table 5. The table is a confusion matrix where the rows are the ground truth classification of the test video sequences, while the rows are the results obtained with the trained neural tree.

		Predicted Event Class		
		Person Walking	Car in the Parking Lot	Bus at the Bus Stop
Ground Truth	Person Walking	47	2	1
	Car in the Parking Lot	0	18	2
	Bus at the Bus Stop	1	2	7

From the table we can see that the neural tree recognized the “person walking on pedestrian paths” with a success rate of 94%, the “car entering/leaving the parking lot” with a success rate of 90%, and the “bus stopping/starting at the bus stop” with a success rate of 70%. The lower rate on the bus stop class can be motivated by the small number of training and test sequences, since during the acquisition time range of 6 hours the number of buses passing in the monitored environment is significantly lower than the number of detected cars and people.

6 Conclusions

Within the current state of the art one of the major problems that afflict the automatic systems of video surveillance is their ineffectiveness in recognizing the time interval in which an event occurs. Moreover, the actual systems are not oriented in distinguishing the possible different characters of a same event. The proposed paper faces the mentioned issues by means of a novel event learning process. The preliminary tests carried out on a set of videos related to a reference outdoor scenario have proven the implemented strategy.

As future works, we are currently evaluating the performance of our approach on more complex and challenging datasets. We are also investigating re-identification [28,29] methods allowing us to recognize events that span large time windows and wide areas that are only partially covered by sensors.

References

1. Martinel, N., Micheloni, C., Piciarelli, C.: Pre-Emptive camera activation for Video Surveillance HCI. In: Intenational Conference on Image Analysis and Processing, Ravenna, RA, pp. 189–198, September 2011
2. Martinel, N., Micheloni, C., Piciarelli, C., Foresti, G.L.: Camera Selection for Adaptive Human-Computer Interface. IEEE Transactions on Systems, Man, and Cybernetics: Systems **44**(5), 653–664 (2014)
3. Martinel, N., Micheloni, C.: Sparse matching of random patches for person Re-identification. In: International Conference on Distributed Smart Cameras (2014)
4. Martinel, N., Micheloni, C.: Classification of Local Eigen-Dissimilarities for Person Re-Identification. IEEE Signal Processing Letters **22**(4), 455–459 (2015)
5. Piciarelli, C., Micheloni, C., Martinel, N., Vernier, M., Foresti, G.L.: Outdoor environment monitoring with unmanned aerial vehicles. In: International Conference on Image Analysis and Processing (2013)

6. Fookes, C., Denman, S., Lakemond, R., Ryan, D., Sridharan, S., Piccardi, M.: Semi-supervised intelligent surveillance system for secure environments. In: IEEE International Symposium on Industrial Electronics, pp. 2815–2820 (2010)
7. Zhong, H.Z.H., Shi, J.S.J., Visontai, M.: Detecting unusual activity in video. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 2 (2004)
8. Zhao, B., Fei-Fei, L., Xing, E.P.: Online detection of unusual events in videos via dynamic sparse coding. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3313–3320 (2011)
9. Xu, J., Denman, S., Fookes, C., Sridharan, S.: Unusual event detection in crowded scenes using bag of LBPs in spatio-temporal patches. In: Proceedings - 2011 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2011, pp. 549–554 (2011)
10. Ghanem, N., DeMenthon, D., Doermann, D., Davis, L.: Representation and recognition of events in surveillance video using petri nets. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop (2004)
11. Nevatia, R., Hobbs, J., Bolles, B.: An ontology for video event representation. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop (2004)
12. Hakeem, A., Shah, M.: Learning, detection and representation of multi-agent events in videos. Artificial Intelligence **171**(8–9), 586–605 (2007)
13. Lin, L., Gong, H., Li, L., Wang, L.: Semantic event representation and recognition using syntactic attribute graph grammar. Pattern Recognition Letters **30**(2), 180–186 (2009)
14. Vernier, M., Martinel, N., Micheloni, C., Foresti, G.L.: Remote feature learning for mobile Re-identification. In: International Conference on Distributed Smart Cameras, pp. 1–6. Palm Springs, CA, IEEE, October 2013
15. Piciarelli, C., Micheloni, C., Foresti, G.L.: Trajectory-Based Anomalous Event Detection. IEEE Transactions on Circuits and Systems for Video Technology **18**(11), 1544–1554 (2008)
16. Micheloni, C., Snidaro, L., Foresti, G.L.: Exploiting Temporal Statistics for Events Analysis and Understanding. Image and Vision Computing **27**(10), 1459–1469 (2009)
17. Agarwal, C., Sharma, A.: Image understanding using decision tree based machine learning. In: ICIMU 2011 : Proceedings of the 5th international Conference on Information Technology & Multimedia, pp. 1–8 (2011)
18. Fischer, Y., Beyerer, J.: Defining dynamic Bayesian networks for probabilistic situation assessment. In: International Conference on Information Fusion, pp. 888–895 (2012)
19. Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C.C., Lee, J.T., Mukherjee, S., Aggarwal, J.K., Lee, H., Davis, L., Swears, E., Wang, X., Ji, Q., Reddy, K., Shah, M., Vondrick, C., Pirsiavash, H., Ramanan, D., Yuen, J., Torralba, A., Song, B., Fong, A., Roy-Chowdhury, A., Desai, M.: A large-scale benchmark dataset for event recognition in surveillance video. In: International Conference on Computer Vision and Pattern Recognition, pp. 3153–3160 (2011)
20. Veeraraghavan, H., Papanikolopoulos, N., Schrater, P.: Learning dynamic event descriptions in image sequences. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
21. Joo, S.W., Chellappa, R.: Attribute grammar-based event recognition and anomaly detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2006 (2006)

22. Moore, D., Essa, I.: Recognizing multitasked activities from video using stochastic context-free grammar. In: AAAI National Conf. on AI, pp. 770–776 (2002)
23. Martinel, N., Micheloni, C., Foresti, G.L.: Robust Painting Recognition and Registration for Mobile Augmented Reality. IEEE Signal Processing Letters **20**(11), 1022–1025 (2013)
24. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: International Conference on Computer Vision, Ieee, pp. 1–8 (2007)
25. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: International Conference on Computer Vision and Pattern Recognition (CVPR) vol. 2, pp. 2169–2178 (2006)
26. Foresti, G.L., Dolso, T.: Adaptive High-Order Neural Trees for Pattern Recognition. IEEE Transactions on System, Man and Cybernetics Part B **34**(2), 988–996 (2004)
27. Piciarelli, C., Micheloni, C., Foresti, G.L.: PTZ Camera Network Reconfiguration. In: Third ACM/IEEE International Conference on Distributed Smart Cameras, Como, Italy (2009)
28. Martinel, N., Micheloni, C., Piciarelli, C.: Distributed Signature Fusion for Person Re-identification. In: International Conference on Distributed Smart Cameras, Hong Kong, pp. 1–6 (2012)
29. Garcia, J., Martinel, N., Foresti, G.L., Gardel, A., Micheloni, C.: Person orientation and feature distances boost Re-identification. In: International Conference on Pattern Recognition (2014)