Exploiting Temporal Statistics for Events Analysis and Understanding

Christian Micheloni, Lauro Snidaro, and Gian Luca Foresti

Department of Mathematics and Computer Science
University of Udine, Italy

Abstract

In this paper, we propose a technique for detecting possible events in outdoor areas monitored by a video surveillance system. In particular, here we focus on the time spent by an object to carry out simple events. To have a statistical representation of the times commonly required to perform certain activities, mixtures of Gaussians are maintained for each event type. Such statistics are then exploited both for the analysis of the simple activities and for discovering anomalous situations (i.e. complex events). In these cases, the system requires the attention of the human operator. A novel way of presenting results to the operator is also discussed. Experiments have been performed on a multi-camera system for parking lot security.

Key words: Event analysis, Event understanding, Video Surveillance, security.

1 Introduction

Research on automatic video surveillance systems is continuously evolving along with the advances in computer vision. Signal and image processing techniques are constantly being updated and improved, while more capable sensors and communication means are being developed [1].

These technical and algorithmic advances are geared towards in increasing accuracy and robustness especially for the low level processing tasks such as target detection and tracking. This is extremely important since higher level analysis modules for behaviour understanding and situation assessment rely on the output from the underlying computer vision algorithms. Although low level processing advances are really important since real scenario often cause even state-of-the-art computer vision algorithms to fail, the high level analysis should receive a bigger attention from the research community.

As matter of fact, scene analysis and understanding are important modules of a visual surveillance system since these modules replicate the knowledge of recognizing anomalous (and potentially dangerous) events. The more intelligence is embedded in the system the less human intervention is required, thus relieving the operator from the chore of continuously observing the video streams produced by the cameras. Different environments have different surveillance requirements ranging from simple motion detection to recognition of complex events. While many available systems claim the capability of performing good motion detection and target tracking, few address the detection of anomalies occurring in the monitored environment.

An anomalous event can be automatically detected as a deviation from common patterns of activity. Then how to represent and manage the knowledge about normal behaviour is a crucial point for behaviour and scene understanding. One way to infer the behaviour of the observed objects is by analysing their movements.

Research on complex event recognition has slowly gained momentum in the past ten years. A brief survey of recent works on the subject can be found in [2]. In addition to that survey, the following works are worth mentioning. Rota and Thonnat [3] proposed a generic framework for real-time interpretation of real world scenes with humans. More recently, an unsupervised technique for detecting unusual events in a large video set was presented in [4], while a solution for tailgating detection at the entrance of a parking lot is described in [5].

The approaches proposed in the literature can essentially be divided in two categories according to the way events are modelled: implicitly or explicitly. In the former ones, no a priori knowledge about the domain is provided. The system automatically identifies common patterns of activity from observed data. A framework for learning such patterns from targets trajectories in a multi-camera system is discussed in [6]. In [7], in addition to trajectories, other features and unsupervised multidimensional clustering of observed data are discussed. These works describe a bottom-up approach that infers high level knowledge from observations.

All the systems that require the explicit definition of what constitutes normal and irregular activities [8] fall in the other category. In this case, the system tries to match the a priori knowledge provided by the operator to observed data patterns. Implicit modelling makes the system highly adaptable to different scenarios and situations, but inaccurate in detecting specific and complex events. On the other hand, explicit modelling generally yields better results in terms of false alarms and missed alarms, but, of course, this method is not self-adapting as all the knowledge is provided by the operator.

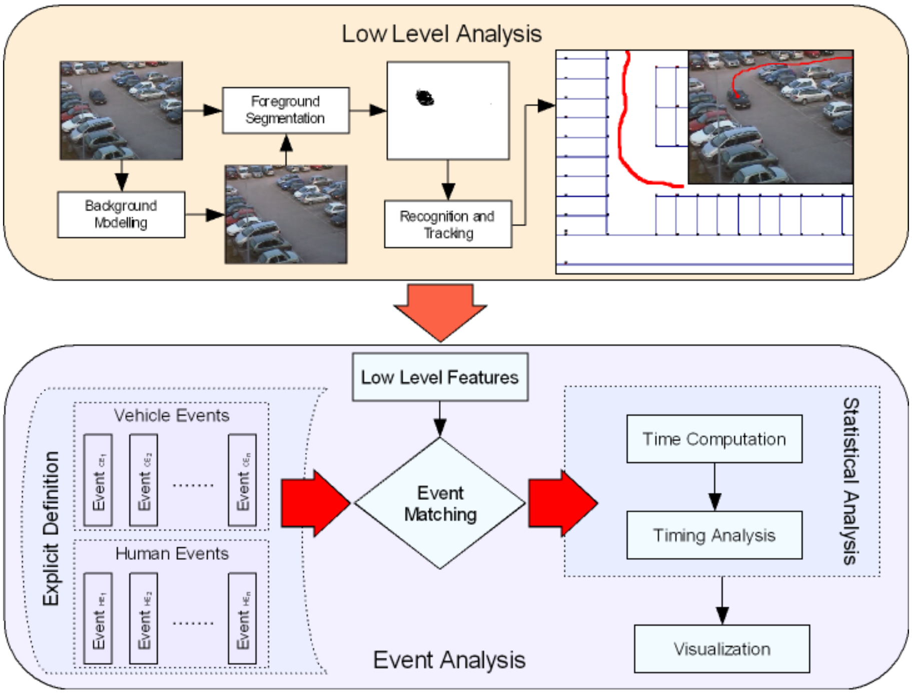


Fig. 1. Logical system architecture

In this latter category, the VERL language [9] constitutes a potentially interesting choice for event representation, however there exists only a partial implementation of it that defines some of the syntactic elements of VERL using the Web Ontology Language (OWL), however, there is no mechanism that defines complex events and no mention on how they should be derived from instances of simple events.

The VidMAP system, presented in [10], is a video-surveillance system employing a high level module that continuously checks if any of the rules defined a priori by an operator is verified. The rules are encoded in Prolog, and a Prolog reasoning engine is used to recognize programmed events.

Regardless of the way the event models are built, automatically from data or manually provided by an operator, the process of recognizing an event is also a debated topic in the literature. Examples are given by [11] and more recently by [12], where a framework for recognizing events by employing a hierarchy of the temporal constraint graph of the models (scenarios). The proposed approches are geared to reduce the computational complexity of the event recognition process, particulary in the multi-target case.

In this paper, we extend the approach proposed by Micheloni *et al.* [13] able to learn basic statistics about the events ongoing in the monitored environment. We propose the analysis of complex events by timing all the simpler events composing them. In particular, the duration times for each type of simple events are described by a distribution that is continuously updated. Events recognized as outliers are flagged as potentially anomalous, thus capturing the attention of the system that eventually can warns the operator. The proposed approach has been tested in an application for video surveillance of a parking lot. The system uses static and moving cameras in a master/slave fashion in order to keep track of the ongoing activities. The active cameras (e.g. PTZ) are controlled by the system to zoom on vehicles entering end exiting the parking area for the identification of the licence plate and its recognition [14].

We also propose a new way to visualize results to the operator so that anomalous events may be easily identified on the screen. In particular, a means for quick visualization of the objects remaining in the scene beyond average time is presented here. Experimental results, performed on real outdoor scenes, have demonstrated the effectiveness of the proposed approach and the added value provided to the operator.

2 System Architecture

The proposed solution is based on a multisensor architecture in which static cameras are used to detect moving objects and to extract information for behaviour understanding purposes, while active cameras (e.g. Pant-Tilt-Zoom PTZ) are exploited for investigating suspicious events and for identifying vehicles and humans. The first level of computation concerns the extraction of moving objects (blobs) from video streams and the computation of relevant features for event analysis purposes (see Fig. 1). Moving objects are detected from the background (background modelling and foreground segmentation) and their movements tracked on a 2D top-view map of the monitored scene [1]. Tracking is performed using a Kalman Filter applied on map positions together with a Meanshift based tracking technique on the image plane [15].

At each time instant, for each object, a set of low level features is maintained. These can be divided in instance and temporal features. The former ones (i.e. object classification, position, dimensions, etc.) are related to the current time instant, while temporal features (i.e. trajectories, mean speed, mean size, etc.) are computed over a time interval. In the proposed approach, events are detected by analysing both instance and temporal features. To obtain a camera geometry independent system , all the kinematic features are extracted on the data computed on the 2D top-view map. This relaxes the problem to a first calibration of the camera for the homography determination. The object classification module distinguishes each detected object within a predefined set of categories (e.g. cars, pedestrians, and groups of people). In the proposed system, an adaptive high order neural tree (AHNT) classifier [16] has been employed.

3 Events Analysis and understanding

Event recognition exploits the output of low level processing modules for target detection and tracking. For each target, positional, temporal, and ID information is used for high-level semantic interpretation of the activities in the monitored scene. The first problem is to choose a suitable way to represent behaviours. There are two main different approaches: a) Probabilistic definition and b) Explicit definition.

In the probabilistic approach, an anomaly is an unusual (infrequent) event. The identification of *normal* patterns therefore leads to the detection of anomalies. The modelling of normal patterns involves a statistical approach in order to discriminate what is normal from what is not. The explicit approach instead is based on the system having a complete description of all the detectable dangerous events. Events detected in the monitored scene are thus matched with those stored in the database of dangerous events. The main disadvantage of this approach is the need for a-priori knowledge of the activities in the application domain.

Furthermore, building the database is not a trivial task. Describing all dangerous events needs to derive all the possible temporal and spatial association of simple events. Thus, even the use of automatic procedure could lead to a difficult solution.

In the next section, the internal representation of simple events, their detection and the techniques for building more complex and more semantically relevant events will be described. Here, an approach based on explicit modelling of dangerous events is discussed. This can be used in combination with the probabilistic approach to achieve more robust results.

Two different types of events have been considered: simple events, characterized by the motion (and behaviour) of a single object (e.g., vehicles, pedestrians, etc. moving in the monitored environment) and composite events, characterized by interactions among multiple objects. A composite event is therefore a complex event generated by a set of temporally consecutive simple events or an event composed by multiple moving objects, e.g., a group of people, a queue of cars, etc.

Simple Events: In an urban environment, a simple event is normally represented by a vehicle, a bus, a motorcycle or a pedestrian moving in the monitored area. A simple event v is defined over a temporal interval $[T^s, T^f]$ and contains a set of features $F = \{f_1, \ldots, f_m\}$ belonging to a given object O_j observed over a sequence of n consecutive frames as:

$$v(T^s, T^f) = \{f_k | f_k \in O_j, k \in [1..m]\} \tag{1}$$

Examples of features are the ID (i.e. person identification or license plate), the class of the detected object, the trajectory, the average speed, the blob shape descriptors, the colour histograms, etc. In the context of a parking lot, examples of simple events can be given by:

- a vehicle enters the parking area
- the vehicle moves with a given trajectory
- the vehicle stops in a given position
- a person exits from the vehicle
- the person moves in the parking area
- the person exits the parking area.

Hence, simple events are generally actions performed within a small area of the monitored environment (entrances, exits, neighbours of a parked car, etc.) or actions characterized by a uniform motion (move, stand, etc.). The main characteristic of such events is therefore their simple and fast detection. This yields to consider them as the building blocks of the scene understanding process.

Composite Events: Composite events are represented by a set of simple events that are spatially and/or temporally correlated. Hence, a composite event is defined over a wide temporal interval as a graph $G(V, E)$ where the set of vertexes $V = \left\{v_1(T^s, T^f), \ldots, v_n(T^s, T^f)\right\}$ is the set of simple events and the set of edges E is the set of the temporal and spatial associations between simple events. In Fig. 2, an example of simple events association is shown. The exploitation of graphs as data structures for representing composite events, theoretically, allows to connect each node (simple event) with any other node in the graph. This can be done regardless of the semantic meaning of the composite event. Instead, even though within the monitored area different spatially related simple events can simultaneously occur, only few of them have significance. For example, it could happen that two cars intersect in a zone inside the map by moving in different directions. In this case, both temporal and spatial correlation exist. Though, a composite event given by two cars moving in opposite directions is not of clear significance.

For this reason, in the proposed solution we restrict the event association to a set of compatible simple events. This is achieved by exploiting a *Event Correlation Diagram* (ECD) that describes the allowed relations between object types, their states and actions. It therefore defines the possible links between different simple events, even when they are generated by different objects. To generate the ECD, the explicitly defined simple events are considered. For each of these, its possible relations with any other defined simple event are analysed and, if any exists, a link between the two simple events is added in the ECD. Hence, building the ECD is an off-line job that is performed by human operators in consequence of the activity knowledge they have about

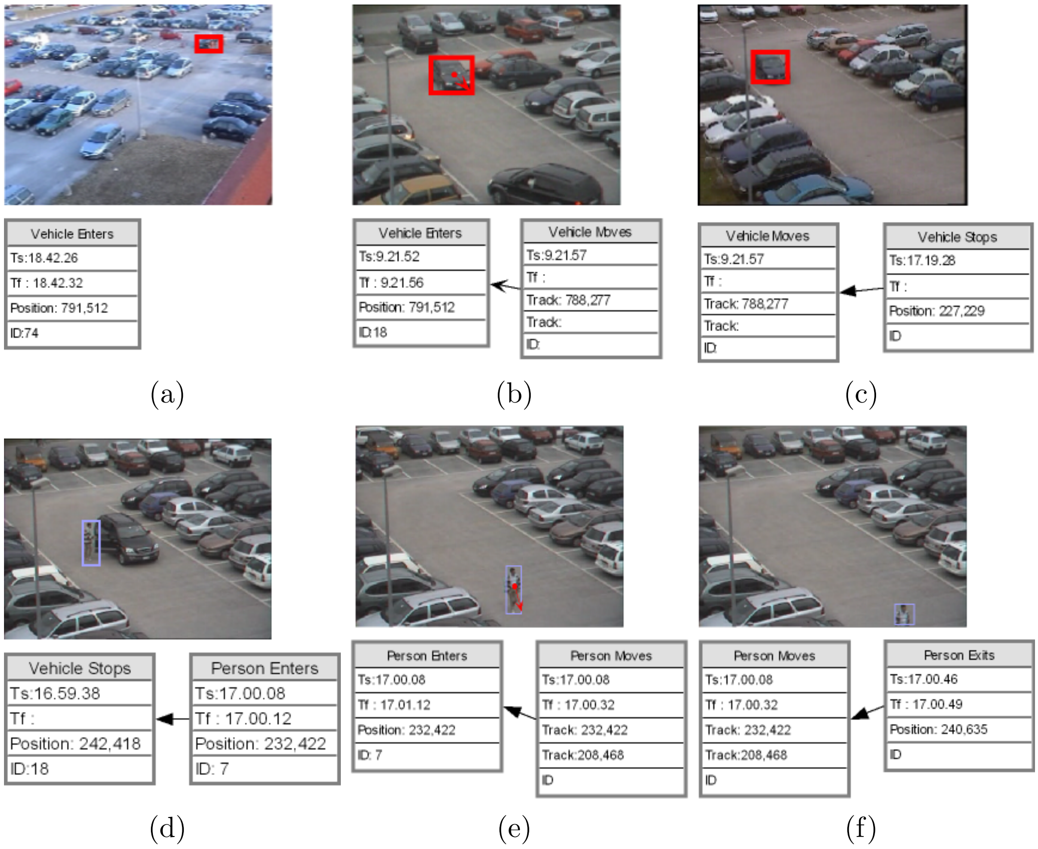


Fig. 2. Examples of some composite events built exploiting the temporal and spatial correlation of two or more simple events. In (a) a node given by a simple event is presented. In (b)-(f) examples of graphs represented by complex events defined by two simple events spatially and temporally correlated. It is worth noticing how in (d) two simple events related to different objects can define a composite event (i.e. person getting out of a car).

the monitored environment.

As shown in Fig. 3 the definition of the admitted correlations between simple events are defined in the depicted ECD. For each event, several characteristic properties (i.e. time of occurrence, position, etc.) are associated to the object that caused the event.

The use of only one out-edge associate to each simple event states a unique available correlation for such an event. Even though in the analysed scenario such a representation has been proven to be sufficient for the majority of the events, there could exist other scenarios requiring a more complex representation. In case that multiple associations are required, instead of increasing the number of out-edges it is possible to split such events in more atomic events. For example, to describe a group of people walking we could describe such an event by describing the single persons walking thus using different simple events. Once the ECD has been defined, its links are used to build the graphs

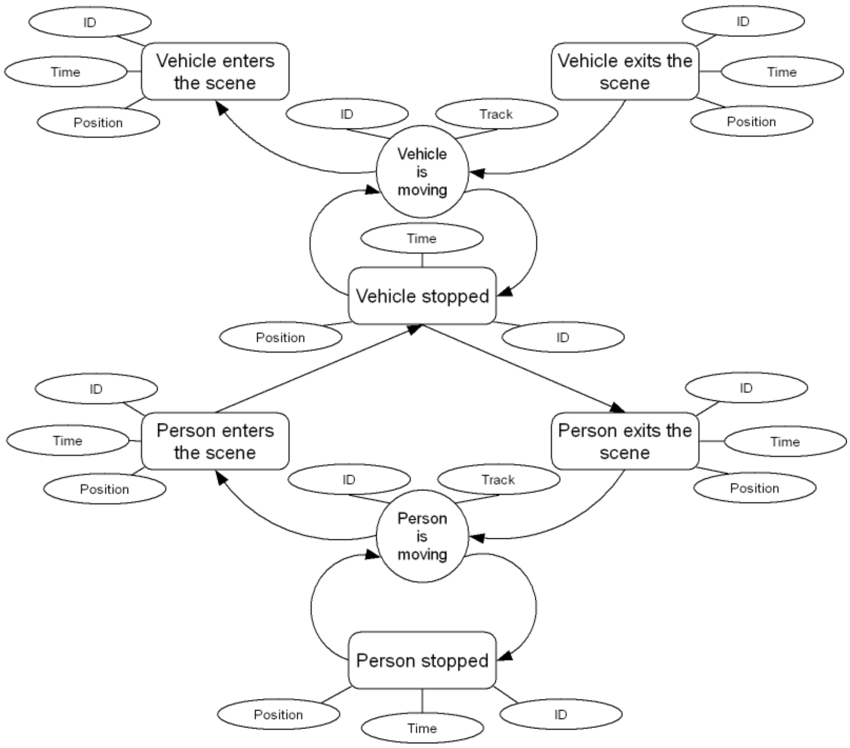
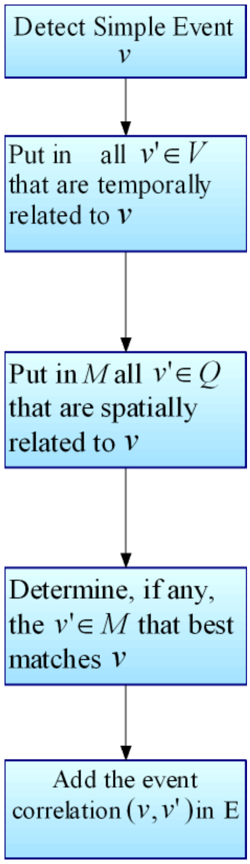


Fig. 3. *Event Correlation Diagram* for composite event detection. Simple events are divided into vehicle and person related events. Events can be correlated to build a composite event if there exists a link between them in the ECD. Person related events can be associated to vehicles events (i.e. person getting in/out of a car) only when it appears/disappears in the neighbourhood of a motionless vehicle.

representing the complex events.

In the chosen test domain (parking lots), normal events are represented by pedestrians walking with typical trajectories (e.g., almost rectilinear for a long number of frames) or vehicles moving and/or stopping in allowed areas. Suspicious events are represented by pedestrians jaywalking, or stopping in a given area for extended periods of time, or loitering around vehicles. Dangerous events are represented by pedestrians or vehicles moving or stopping in not allowed areas. An off-line Event Graph (EG) is built with the models of suspicious simple/complex events. The EG is a graph where single unconnected nodes represent simple suspicious events (i.e. person moving with a particular trajectory) while connected nodes represents composite suspicious events. As shown in the algorithm of Table 1, for each new event a node with related information/features representing the motion or the characteristics of a given object is extracted and stored in the Active Event Graph (AEG). The AEG is therefore updated every time a new simple event has been detected. To reduce the graph growth, old events (i.e. events whose ending time is older than a predefined threshold) are pruned to expedite further searches. Determining simple suspicious events is carried out in two phases: a) a neural network classification and b) graph similarity.



$v(T^s, T^f)$=Detect Event()
Insert $v(T^s, T^f)$ into $G(V, E)$
for all $v' \in V$ **do**
  if v' and v are temporal related **then**
    Insert v' into TimeEvents
  end if
end for
for all $v' \in TimeEvents$ **do**
  if v' and v are spatially related **then**
    Insert v' into Events
  end if
end for
match=NULL
for all $v' \in$ Events **do**
  if ECD allows (v', v) **then**
    if v' is more probable than match **then**
      match=v'
    end if
  end if
end for
if match neq NULL **then**
  Insert $(match, v)$ into E
end if

Table 1
The left chart shows the flow diagram of the main steps adopted to correspond temporally and spatially the events of interest. A deeper representation is given on the right by including a pseudo-code of such an algorithm.

In the first phase, the object trajectory is smoothed by a polynomial curve fitting procedure that exploits the least-squares method for the computation of a Bezier-Spline. Trajectories are event independent and the degree of the smoothing polynomial is object dependent. In our solution, since vehicles due to their steering properties have lower motion abilities than humans, we adopted a lower degree for vehicles and an higher degree for people (e.g. third and fifth degree respectively for cars and humans). The m coefficients returned as output of such a procedure are therefore given in input to the Adaptive High Order Neural Tree [16]. Such a tree has been previously trained with the suspicious simple events present in the EG and with a set of normal simple events. The output is a binary classification among normal and suspicious events. If the current event is classified as suspicious it is directly signalled to a human operator.

On those events that have been classified as normal, a second phase is executed. That is, their features are compared with those of the events present in the EG. For example, a person walking in a forbidden area, a car moving in a prohibited direction etc. Even in this case if a similarity is found, the anomaly is directly signalled to a human operator.

After this first step, all detected events are further analysed in order to identify potential anomalous composite events (see diagram and related algorithm in Table 1). Every time a new simple event is added to the AEG, this is inspected to find some correlations between the current event and those previously detected. A first search is performed according to the the time property associated to each event. In particular, for each event v the time property is composed by the starting T_v^s and ending T_v^f time attributes. Hence, first of all, the starting time T_v^s of the current event v is compared with the ending times $T_{v'}^f$ of the most recent events v' into the AEG. The difference between the two times is thresholded to find a temporal matching.

Event correlation is primarily based on temporal matching since the detection of anomalous events that are not temporally related is demanded to the explicit definition of such anomalous events. For the sake of explanation, we may think to an event concerning a parked car . Let us now suppose that the driver gets off the car after a long time. The temporal correlation fails and the event "*Person Enters*" is not linked to the parked car. However, the explicit definition represented in the EG takes care of this case by adding such a composite event in the AEG. Hence, the alarm is raised as soon as the temporal threshold for the correlation of "car stopped" and "person enters" event is passed.

Once the first matching process is completed, a new phase investigates about spatial correlations between the current event and the events that have been temporally matched. The ending position $P_{v'}(x, y)$ of each temporal correlated event is compared with respect to the starting position $P_v(x, y)$ of the current event. If the distance between the two positions falls below an experimentally defined threshold, then the two events can be composed together. If both temporal and spatial properties agree and if the matching is allowed by the finite automaton, a new event, virtually given by the concatenation of simple events, is created.

4 Timing Analysis

Time can be used as a key feature to distinguish between usual and unusual activities. Let us think to the time required for a person to get out the car after having parked. Too short a time could hint the necessity to escape from the parking spot as fast as possible, whereas too long a time could be related to a person who is checking up on a possible target. Therefore, looking for time patterns in activities boils down to building probability distributions of the time needed to complete a certain action.

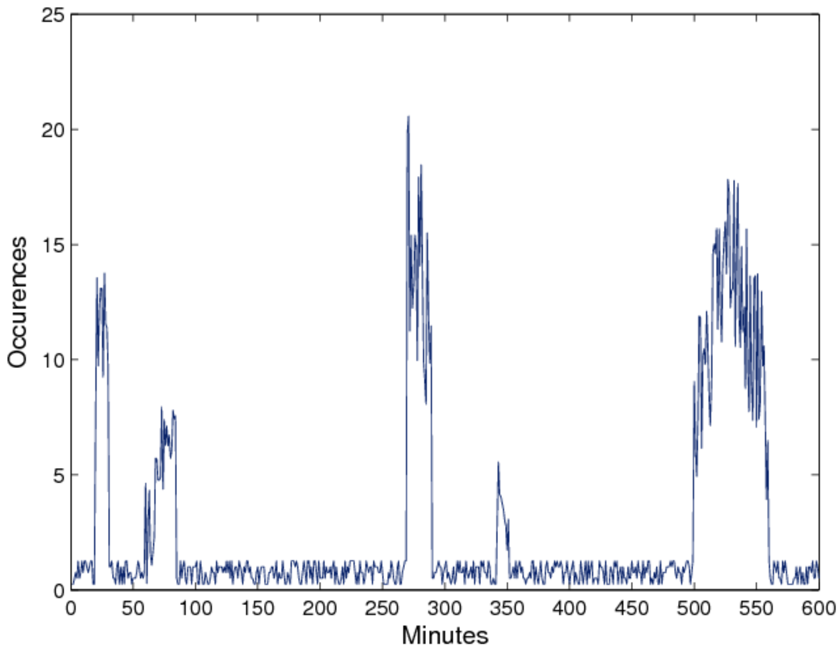


Fig. 4. Example of computation of parking times. The chart plots computed times in a 5 days analysis from 8.30 A.M. to 6.30 P.M.

Hence, it is by computing the time requested to perform simple and complex events that the system is able to build a probabilistic representation of which timings are usual inside the monitored environment. In Fig. 4 an example of the time computation is shown. Let $\{T_1^e, \ldots, T_n^e\}$ be the history of the last n computed timings of an event e, the corresponding multimodal histogram suggests a representation described by means of a mixture of narrow Gaussians. In particular, to describe the probability distribution of the timing history of each event we adopted a mixture of M Gaussians where the probability to observe a duration T of an event e is described by:

$$P(T^e) = \sum_{i=1}^{M} \omega_i * G\left(T, \mu_i, \sigma_i\right) \tag{2}$$

where M is the number of Gaussian distributions in the mixture, ω_i is the weight given to the $i - th$ Gaussian, and μ_i and σ_i are its mean and its standard deviation respectively. G is the Gaussian probability function given by:

$$G\left(T, \mu_i, \sigma_i\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(T-\mu)^2}{2\sigma^2}} \tag{3}$$

With such a formulation we have that each event timing is described by a mixture of Gaussians where the current value is represented by either one of the Gaussian or none. In the former case, the current value is used to update the model, while in the latter is used to highlight an unusual behaviour. Thereafter, if the higher level modules of the systems assert that the current event is not anomalous, the new value is used to update the model. A time value is represented by the mixture if there exists a Gaussian that contains the value. Hence, if it is within $k\sigma$ from the mean μ of one of the M Gaussians, where k is a per event/per Gaussian experimentally defined threshold, the mixture of Guassians is updated:

$$\omega_i = \begin{cases} (1-\alpha)\omega_i + \alpha & \text{if the } i-th \text{ distribution contains} \\ & \text{the value} \\ (1-\alpha)\omega_i & \text{otherwise} \end{cases} \tag{4}$$

where α is the learning rate. In addition, the $i-th$ distribution is updated as follows:

$$\mu = (1-\rho)\mu + \rho T \tag{5}$$

$$\sigma^2 = (1-\rho)\sigma^2 + \rho(T-\mu)^2 \tag{6}$$

where ρ is the learning factor.

5 Complex events classification

Recognizing composite events means finding a temporally ordered sequence of simple events following some predefined pattern. To accomplish this, we employ graph matching. Graph matching is a classical optimization approach which has been studied for a number of years and applied in many disciplines such as pattern recognition and computer vision [17]. Classical concepts are graph and subgraph isomorphisms. If an isomorphism can be established, that is, a bijection transforming a graph G_1 into a graph G_2 and vice versa, then the two graphs are the same. If one of the two graphs, say G_1, is larger than the other, G_2, then it has to be matched against subgraphs of G_1.

In particular, it is often the case where a template graph, which describes some a priori information of interest drawn from the knowledge base, has to be matched with a data graph that represents the information collected from the sensors. The process is geared to recognizing occurrences of modelled information in the sensory data. Here, we employ this approach to recognize sequences of simple events constituting complex events of interest among detected sequences of correlated (temporally and spatially) simple events.

Of course, real-world conditions, the bijective condition is generally too strong. That is, detected sequences of simple events can differ from templates due to a number of reasons. For example, Figure 5(b) shows three temporally consecutive simple events that can constitute the complex event "Car arrives and parks" if the stopping position is a valid parking position. The graph in Figure 5(a) shows a data graph describing the behaviour of a car entering the scene, stopping while looking for a spot, moving again, and finally stopping in its final parking position. Therefore, the recognition process calls for an

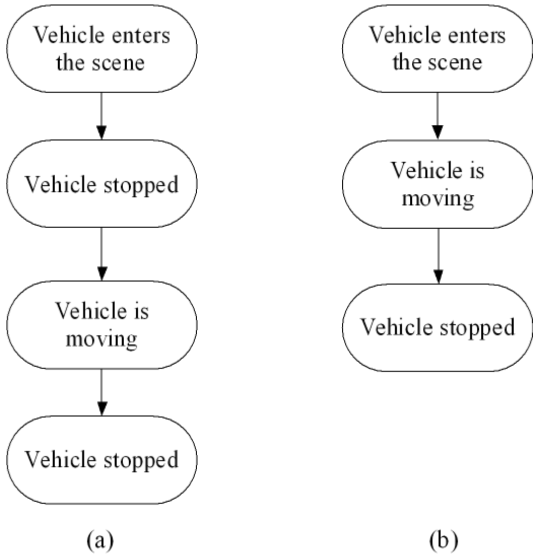


Fig. 5. Example of graphs. (a) Data graph representing detected behaviour and (b) template graph of a parking sequence.

inexact matching algorithm [18–20]. This kind of weaker matching is called "homeomorphism", which drops the conditions that nodes in the first graph have to be mapped to distinct nodes of the second graph.

We employ attributed relational graphs, where a vector of features is associated to each node. Let $G = (N, E, \mu)$ denotes a graph, where N is the set of vertices, $E \subseteq N \times N$ is the set of edges, and $\mu : N \rightarrow \mathbb{R}^m$ where m is the number of node attributes. A solution to the inexact matching problem can be expressed as a subgraph $G_S = (N_S, E_S)$ of the the association graph $G_A = (N_1 \times N_2, E_A)$ between $G_1 = (N_1, E_1)$ and $G_2 = (N_2, E_2)$, such that $\forall a_1 \in N_1, \exists a_2 \in N_2, (a_1, a_2) \in N_S$ and $\forall (a_1, a_2) \in N_S, \forall (a'_1, a'_2) \in N_S, a1 = a'_1 \Rightarrow a_2 = a'_2$.

As described in [18], the matching is here performed by minimizing the following function:

$$f(G_S) = \frac{\alpha}{|N_S|} \sum_{(a_1,a_2)\in N_S} c_N(a_1, a_2) + \frac{(1-\alpha)}{E_S} \sum_{e\in E_S} c_E(e) \tag{7}$$

where $c_N(a_1, a_2)$ is a dissimilarity function between the attributes of a_1 and a_2. In practice, the node attributes employed in the experiments were time and 2D position on the map, thus accounting for three features. c_N was defined as the Euclidean distance between those features. The α parameter was set to .5, and the edge dissimilarity measure employed in [18] was employed for c_E. In addition, to ease the process, composite events within each category are simplified. Only entering and exiting events with their temporal properties are considered. Moreover, if two different objects generate a complex event also the events that relate the two objects are considered. For example composite events related to single objects become {{*car enters, car exits*} or {*person enters, person exits*}, while in case of two objects interacting they can be summarized by {*car enters, car stops, person enters, person exits, car stops, person enters*}}.

6 Experimental Results

To test the proposed solution we have run the system for a period of time of about 10 hours. In particular, we have used two static cameras to analyse the entire parking lot area in front of our University building and a Pan-Tilt-Zoom (PTZ) camera for close-up acquisition of targets involved in suspicious events. The PTZ is calibrated such that by providing a position on a 2D top view map it is possible to define pan and tilt angles in order to get the point on the map at the centre of the PTZ image. The zoom factor is selected on the basis of how far the object from the camera is and the size the object has on the static cameras.

During the experimentation phase the system has detected 942 simple events. Of these, 56 were related to false events principally due to errors in low level task (i.e. motion detection, tracking, etc.). Of the remaining 886 simple events, 824 have been correlated to other events while just 62 events have not been correlated to any event. This last value represents an error of the system since from the definition of our simple events any object entering the scene generates at least two related events (entrance and exit).

Of the remaining complex events the system has correctly identified the 4 suspicious events that we have simulated on purpose. A first event consisted in a person walking between the parked cars. This type of event was defined as suspicious thinking to a person looking for a car to rob. This event has been correctly detected by the neural network on the person's trajectory.

A second event was about a situation in which a person parks a car in a free spot close to the University building leaving the scene on board of a previously parked car. This event was supposed to represent the placing of a car bomb.

The detection of such an event has been performed by matching the computed event graph with those stored in the EG.

The other two events, even though explicitly defined, have not been detected by applying the neural network or the graph matching approach. However, among the considered events for a parking lot application described in Section 3, a particular attention has been devoted to the time spent by the objects while being stationary.

As discussed in Section 4, statistical information can be collected for each event thus allowing the system to detect outliers as possible anomalous events. In a parking lot, for example, parking times could be of particular interest. A vehicle left parked for days in a public area where everybody is supposed to leave after working hours is probably something that should capture the attention of the system, that in turn should signal the event to the operator. The vehicle could in fact be left abandoned, or, in a defence oriented application geared to protect sensitive buildings, such a vehicle could even constitute a possible terrorist threat. Here, an idea for easy visualization of the vehicles whose parking time can be considered an outlier event is presented. This is purely for facilitating the operator in spotting those vehicles that can eventually be checked: for example, if the organization, who owns the surveillance system, maintains a database of the licence plate numbers of the vehicles belonging to its employees, then suspect vehicles can be immediately identified; if the vehicle doesn't appear to be registered in the database, the event could definitely be considered as something anomalous.

An example of the approach for easily conveying to the operator the location and the parking time of suspicious vehicles, is shown in Fig. 6. The parking time is represented as a column rising from the ground in the spot occupied by the vehicle. The images in the sequence show how the columns rise as time passes. A colour scheme can also be defined to further improve the visualization of outliers. In the figure, the parking time of the car in the bottom left corner is considered an outlier and consequently its column is coloured in red, while the parking time of the car in the upper right corner is k times the standard deviation σ from the mean of the matched Gaussian but it is still not considered an outlier and its column is coloured in blue. The parameter k is chosen by the operator depending on the particular application. The operator could also toggle such visualizations or decide to visualize only outliers. The columns can be used to represent not only parking times, but also every event where an object remains stationary for an amount of time exceeding the mean (i.e. k times the standard deviation as above) the distribution of that particular event. This means that the height of the columns does not represent time in absolute terms, but it is relative to the time distribution of each particular event.

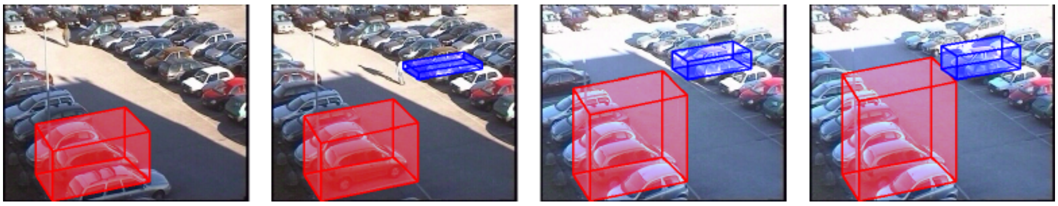


Fig. 6. Visualization of parking times. The system draws columns where vehicles exceeding the mean parking time are located. The height of the columns increases as time passes allowing the operator to quickly inspect the scene for this kind of anomalous event.

To prove the effectiveness of the proposed approach, the system has been executed on sequences belonging to the i-Lids [1] dataset. The chosen test-bed of such sequences is represented by an urban road. This experiment aimed to detect cars parked along the roadside near the curb. The system has been initialised to maintain a single mixture of Gaussians for representing the parking times alongside the curbs. This means that all computed parked times in such an area have to be compared with such a mixture. For this purpose, a grid based mask can be defined to group a set of neighbour pixels. Such a grid is a per event defined and hence its granularity can be a priori decided (i.e. greater for vehicles, smaller for persons,etc.). This allows to describe all the pixels masked by a grid' cell with a unique mixture. In this experiment, we have defined a mask with a unique cell that groups all possible positions where a car can stop near the curb. In Figure 7 the map representing the test-bed area and the parking zone are respectively presented in the two images.

During the experiment we have run the system over a first sequence in which a car remains stationary in the area of interest for $4.87s$. During this analysis there is no Gaussian distribution present in our model. Therefore, the system considers such an event anomalous from the time instant in which the car stops. However, the visualisation system starts to display the alert box only after $3s$ from the anomaly signal. Such a period of time has been introduced to filter out all those possible false positives generated by low level tasks (i.e. change detection, etc.). Afterwards, the event has been manually declared normal affecting the mixture. Therefore, a first Gaussian is added with $\mu = 4.87$ and $\sigma = 1$ considering the effective time elapsed as the mean and a computation error of $1s$ as the standard deviation of the distribution representing such an event. In Figure 8, the key frames representing the event of this first sequences are shown.

With the mixture *trained* with the first sequence a further experiment has been carried out on a new dataset sequence. A first car stops and remains stationary for $4.27s$. Adopting a strategy of considering the events not represented by

[1] Imagery Library for Intelligent Detection Systems (i-LIDS) is the new UK Government standard for Video Based Detection Systems (VBDS)

(a) (b)

Fig. 7. (a) Map representation of the i-Lids test-bed area. (b) The red strip represents the parking area initialized with the same mixture of Gaussians.

(a) (b) (c) (d)

Fig. 8. Results obtained on the sequence rd6c7 of the i-Lids dataset. (a) The system detects a car stopping at time $19,353s$ and frame 580. After $3s$ at frame 670 (b) the system still considers the even normal. Starting from frame 671 (c) the system labels the events as anomalous till time $24.224s$ when the car regain motion (d). The total time elapsed while the car were stationary is of about $4.87s$.

a Gaussian if their duration is far from the mean more than two times the standard deviation, we obtained that this first event is considered normal from the actual mixture as it belongs to the only Gaussian in it. The first row of Figure 9 presents some frames displayed by the visualisation system.

The match of this event with the unique Gaussian in the mixture results in the update of such a Gaussian. Adopting $\rho = 0.3$, to give major weight to the past than to the current measure, the updated distribution, computed by (6), has a mean $\mu = 4.771$ and a standard deviation $\sigma = 0.85$.

A second event, generated by a car parking in the area of interest is detected on the same sequence and analysed by using the updated mixture. In this case, the car remains stationary for a period of time that is longer than $20s$. For the first $6.51s$ the event is considered normal as it matches the Gaussian

Fig. 9. Results obtained on the sequence rd8c7 of the i-Lids dataset. The first row presents key frames of a normal event. From the time instant in which the system detects a car stopping $(11, 245s)$ to the time instant in which it detects the car moving again $(15.516s)$ the total time elapsed has been $4.27s$. At time instant $42.809s$ the system detects a new event related to a car stopping (e). For the first $6.45s$ the system considers the event belonging to the Gaussian present in the mixture (f). From time instant 49.316 (g) till the end of the sequence (h) the system labels the event as anomalous since its duration falls outside all Gaussians in the mixture.

in the mixture ($\mu + 2\sigma = 6.471$). From that instant on the event is flagged as anomalous. This is displayed to the operator who can see an alert box raising higher and higher above the car as time passes.

It is worth noticing how the system, after the first event has been able to discriminate between normal and anomalous events. As matter of fact, in the second sequence the drop off of a passenger is considered normal while parking a car is not.

To test a system on a more challenging scenario, a complex event in a parking lot has been analysed to illustrate the effectiveness of the proposed approach. The sequence in Fig. 10 shows a car approaching the entrance gate of the parking area as detected by the wide-angle static camera (a), this event triggers the active camera that is positioned and zoomed so to acquire the licence plate number of the car (b). The car then proceeds in the parking area looking for a spot (c). Even if a spot is available, the driver decides to stop the car in the middle of the lane and exits from the car (d). The car is left stationary in this forbidden parking position for an excessive time and a column starts to grow in that position to capture the attention of the operator (e). The driver is still loitering around the vehicle as time passes (f). The driver re-enters in the car (g), and drives towards the exit (h). When this last event is detected, the system guides the active camera in order to acquire the licence plate number

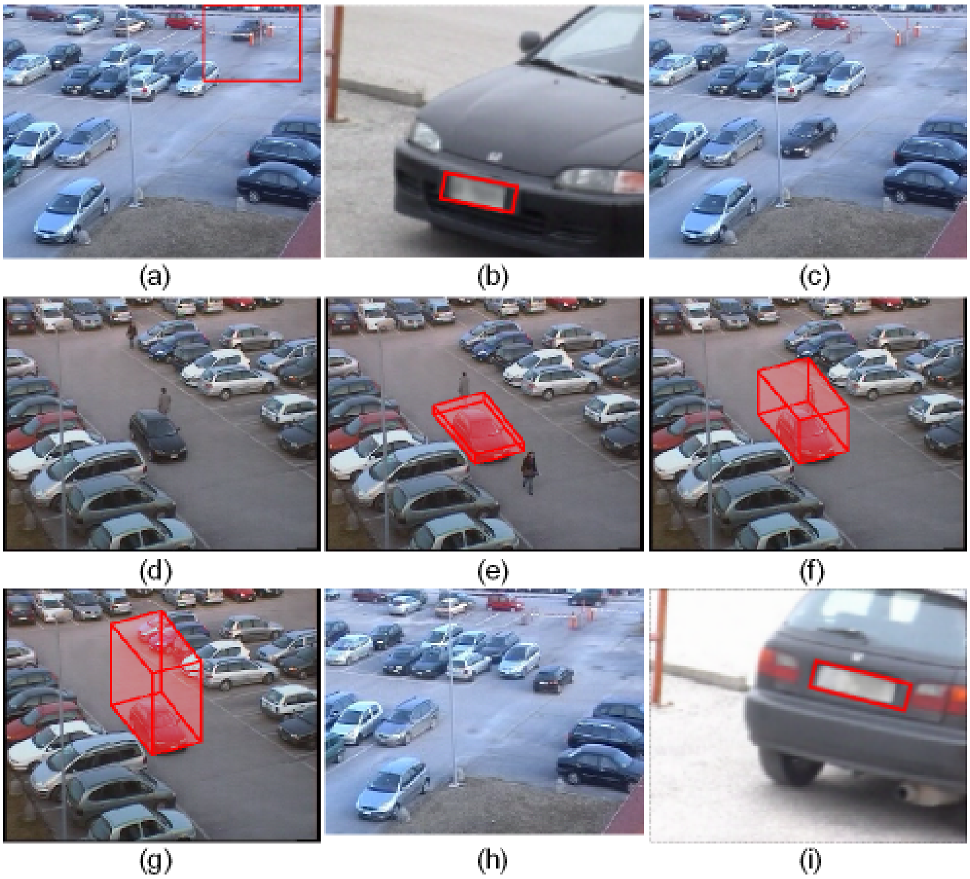


Fig. 10. Anomalous event detection. A car is detected close to the entrance of the parking area (a). Its identification is performed by tasking and active camera (b). The car is tracked (c) until it stops outside a parking spot (d). Statistics are collected about this unusual stationary condition (e-g). Finally, it is tracked on its way towards the gate (h), and its licence place is matched with the previous one (i).

as the car leaves the area (i).

On the scenario described in Fig.10, the proposed system is able to appropriately match the simple events by analysing temporal and spatial properties. As result, it has been able to build the graph of events depicted in Fig 11.

Moreover, in this case the system highlighted two suspicious activities: one related to a person and the other to a vehicle. While the car remains parked out of spot for $3'.11"$, the mean and standard deviation of the available Gaussian distribution for such event are respectively $\mu = 1'.15"$ and $\sigma = 21"$. Having selected $k = 1'.8"$, the anomaly is detected after $T = 1'.52"$. Meanwhile, the driver got out of the car (enters the scene), walked around, and finally got in the car again (exits the scene). Even though the composite event within the person category is considered normal, when it is associated with the parked car it becomes unusual as the statistics generated by a person to get out of and get back in the car have a lower or a larger mean time (i.e 15" and 25').

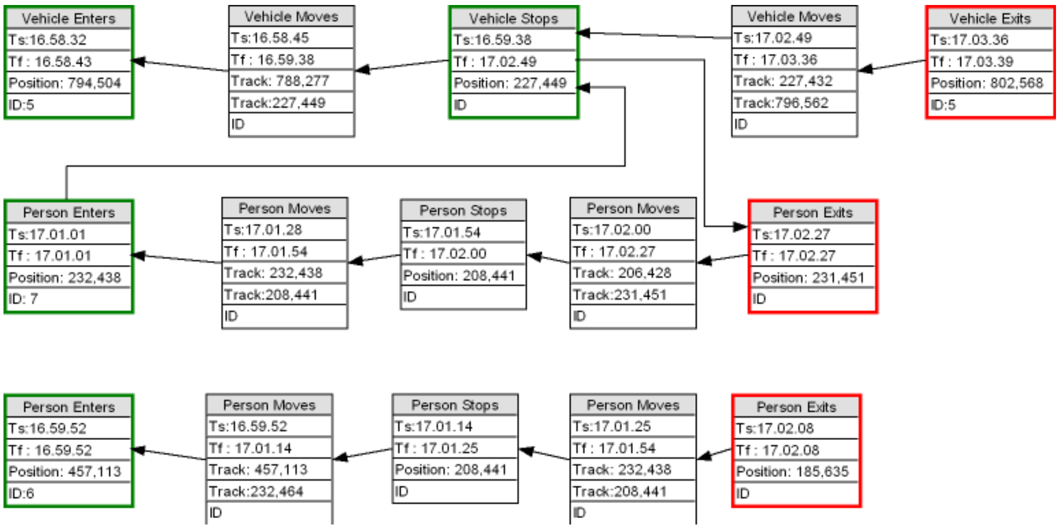


Fig. 11. Example of composite events detection. A set of vehicles events are easily matched by their temporal properties. Concerning person events, it is worth noticing how, thanks to both temporal and spatial properties, it is possible to link the events regarding the person with $ID = 7$ to vehicle events, while those related to person with $ID = 6$ describe a path of a walking person not correlated to any vehicle.

7 Conclusions

The main focus of this paper is statistical event analysis for multi-camera video surveillance. The proposed approach allows the system to keep track of the time needed by the objects to complete simple and composite events. For each of them, a multi-Gaussian distribution is maintained and outliers are detected as possible anomalous events, thus capturing the attention of the system. The proposed approach has been applied to an application for parking lot surveillance. Particular attention has been dedicated to the time spent by objects while being stationary. An idea for quick visualization of the vehicles exceeding the mean parking time, so that the operator could easily inspect the scene, has been presented. Experimental results on real outdoor scenes have proven the effectiveness of the system.

Acknowledgements

This work was partially supported by the Italian Ministry of University and Scientific Research within the framework of the project "Ambient Intelligence: event analysis, sensor recofiguration and multimodal interfaces"(2006-2008) and by the European FP6 Project HAMLET "Hazardous Material Localisation & Person Tracking".

References

[1] G. L. Foresti, C. Micheloni, L. Snidaro, P. Remagnino, T. Ellis, Active video-based surveillance systems, IEEE Signal Processing Magazine 22 (2) (2005) 25–37.

[2] C. Piciarelli, G. L. Foresti, L. Snidaro, Trajectory clustering and its applications for video surveillance, in: Proceedings of the International Conference on Advanced Video and Signal based Surveillance, Como, Italy, 2005, pp. 40–45.

[3] N. Rota, M. Thonnat, Video sequence interpretation for visual surveillance, in: Proceedings of the Third IEEE International Workshop on Visual Surveillance, Washington, DC, USA, 2000, pp. 59–68.

[4] H. Zhong, J. Shi, M. Visontai, Detecting unusual activity in video, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, Washington DC, U.S.A, 2004, pp. 819–826.

[5] G. Diamantopoulos, M. Spann, Event detection for intelligent car park video surveillance, Real-Time Imaging 11 (3) (2005) 233–243.

[6] L. Snidaro, C. Piciarelli, G. L. Foresti, Activity analysis for video security systems, in: Proceedings of the IEEE International Conference on Image Processing, Atanta, GA, USA, 2006, pp. 1753–1756.

[7] F. Porikli, T. Haga, Event detection by eigenvector decomposition using object and frame features, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, pp. 114–121.

[8] Y. A. Ivanov, A. F. Bobick, Recognition of visual activities and interactions by stochastic parsing, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 852–872.

[9] A. R. Francois, R. Nevatia, J. Hobbs, R. C. Bolles, J. Smith, VERL: an ontology framework for representing and annotating video events, IEEE MultiMedia Magazine 12 (4) (2005) 76–86.

[10] V. D. Shet, D. Harwood, L. Davis, Vidmap: video monitoring of activity with prolog, in: Proceedings of the IEEE Advanced Video and Signal Based Surveillance Conference, Como, Italy, 2005, pp. 224–229.

[11] V. Vu, F. Bremond, M. Thonnat, Automatic video interpretation: a novel algorithm for temporal scenario recognition, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence, 2003, pp. 9–15.

[12] C. Dousson, P. L. Maigat, Chronicle recognition improvement using temporal focusing and hierarchization, in: 20th International Joint Conference on Artificial Intelligence (IJCAI07), 2007, pp. 324–329.

[13] C. Micheloni, L. Snidaro, G. L. Foresti, Statistical event analysis for video surveillance, in: Proceedings of the Sixth IEEE Internationl Workshop on Video Surveillance, Graz, Austria, 2006, pp. 81–88.

[14] S.-L. Chang, L.-S. Chen, Y.-C. Chung, S.-W. Chen, Automatic license plate recognition, IEEE Transactions on Intelligent Transportation Systems 5 (1) (2004) 42–53.

[15] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis Machine Intelligence 25 (5) (2003) 564–575.

[16] G. Foresti, T. Dolso, Adaptive high-order neural trees for pattern recognition, IEEE Transactions on System, Man and Cybernetics Part B 34 (2) (2004) 988–996.

[17] D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition, International Journal of Pattern Recognition and Artificial Intelligence 18 (3) (2004) 265–298.

[18] R. M. Cesar-Jr., E. Bengoetxea, I. Bloch, P. Larrañaga, Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms, Pattern Recognition 38 (11) (2005) 2099–2113.

[19] T. Caelli, S. Kosinov, An eigenspace projection clustering method for inexact graph matching., IEEE Trans Pattern Anal Mach Intell 26 (4) (2004) 515–519.

[20] A. Hlaoui, S. Wang, A new algorithm for inexact graph matching, in: Proceedings of the International Conference on Pattern Recognition, Vol. 4, IEEE Computer Society, Los Alamitos, CA, USA, 2002, pp. 180–183.