A semantic-based probabilistic approach for real-time video event recognition[☆]

Juan C. SanMiguel[1,*], José M. Martínez

Video Processing and Understanding Lab, Escuela Politécnica Superior, Universidad Autónoma of Madrid, E-28049, Madrid, Spain

Abstract

This paper presents an approach for real-time video event recognition that combines the accuracy and descriptive capabilities of, respectively, probabilistic and semantic approaches. Based on a state-of-art knowledge representation, we define a methodology for building recognition strategies from event descriptions that consider the uncertainty of the low-level analysis. Then, we efficiently organize such strategies for performing the recognition according to the temporal characteristics of events. In particular, we use Bayesian Networks and probabilistically-extended Petri Nets for recognizing, respectively, simple and complex events. For demonstrating the proposed approach, a framework has been implemented for recognizing human-object interactions in the video monitoring domain. The experimental results show that our approach improves the event recognition performance as compared to the widely used deterministic approach.

Keywords: Video event detection, Semantic video analysis, Bayes Network, Petri Net, Low-level uncertainty

1. Introduction

The recognition of human-related events has recently become a relevant research area motivated by the variety of promising applications such as video surveillance, human-computer interaction and content-based indexing. Moreover, this interest can be also explained by the maturity of the employed low-level tools. Nevertheless, it still presents many challenges such as the uncertainty of the low-level tools (e.g., object detection and tracking), the limited availability of training data, the similar appearance of different events and the modeling of complex relations.

Many approaches have been proposed for event recognition which can be roughly classified into semantic and probabilistic. Semantic (or deterministic) approaches are based on defining rules to model the events [1]. However, current approaches only describe a small portion of semantics (e.g.,

[☆]This work has been partially supported by the Spanish Administration agency CDTI (CENIT-VISION 2007-1007), by the Spanish Government (TEC2011-25995 EventVideo), by the Consejería de Educación of the Comunidad de Madrid and by The European Social Fund.

*Corresponding author

Email addresses: juancarlos.sanmiguel@uam.es (Juan C. SanMiguel), josem.martinez@uam.es (José M. Martínez)

scene layout [2], event definitions [3]), they do not suggest the appropriate recognition strategies and they do not consider the uncertainty inherent to low-level observations and event definitions. On the other hand, the probabilistic approaches have shown a superior performance as compared to the semantic ones [4]. They accurately learn event models from training data achieving high precision within a domain and allowing an intrinsic uncertainty handling. However, they are not able to model complex relations and their usage is limited for different, albeit related, domains. In this situation, a combination of both approaches would be desirable for solving these limitations. Although this combination is gaining attention in the recent years, current approaches are limited to the definition of simple events [5], the assumption of accurate low-level analysis [6] and the use of domain-dependent recognition strategies [7]. Thus, their extension to generic recognition of complex events considering low-level uncertainty is not a straightforward task.

This paper addresses the above-mentioned limitations by introducing a new approach for event recognition that takes advantages of the accuracy of probabilistic approaches as well as the descriptive capabilities of semantic-based approaches. We start from previous work [8] in which Bayesian Networks (BNs) are manually defined for real-time recognition of simple events. We propose a framework for complex event recognition based on hierarchical event descriptions that can be applied to a large variety of domains. This framework extends [8] in three aspects. First, a state-of-art approach is integrated for event representation [9]. The hierarchy of this representation model allows to apply recognition strategies suitable to each event type. Hence, a two-layer structure is defined for recognizing simple and complex events. Simple events are recognized by means of BNs as in [8], but in this work BNs are created automatically. The second extension regards the recognition of complex events by coupling the BNs with probabilistically-extended Petri Nets (PNs). The third extension defines a methodology to convert the event descriptions into their recognition models. Thus, BNs and PNs are built automatically from respectively, simple and complex event descriptions. We demonstrate the validity of the proposed approach for recognizing human-object interactions in the video monitoring domain. Experimental results show that it outperforms the widely used deterministic approach for recognizing events performed by different people in diverse scenarios whilst operating at real-time.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 overviews the framework for video event recognition whilst section 4 describes the two-layer recognition structure. Then, section 5 illustrates its application to the video surveillance domain and section 6 presents some experimental results. Finally, section 7 concludes this paper.

2. Related work

Recently, several surveys have been published focused on the recognition of human-related video events [1][4][10][11]. We build on [4] and discuss the approaches based on Pattern Recognition (PR), Graphical Modeling (GM) and Semantic Modeling (SM). Additionally, Hybrid Modeling (HM) is included to describe combinations of them

2.1. Pattern Recognition approaches

PR approaches recognize events as a traditional classification problem in which few semantic knowledge is needed and therefore, they are simple, well understood and easy to implement. Accurate event models can be learned from training data. There are many examples such as nearest neighbor [12] and neural networks [13]. Their advantages are the automatic learning of event models, the high-precision within a domain and the management of the low-level analysis uncertainty.

On the other hand, they tend to increase the computational complexity, they are not able to model complex spatio-temporal relations and their usage is limited for different, albeit related, domains.

2.2. Graphical Modeling approaches

GM approaches model the spatio-temporal event structure as a sequence of *states* by using semantic knowledge. Their descriptive capability is highly increased as compared to PR approaches. Current literature can be classified into deterministic (DGM) and probabilistic (PGM).

DGM approaches assume fully observable event states and accurate low-level analysis. Their structure is typically specified by expert knowledge. For instance, Finite-State-Machines (FSMs) have been proposed for modeling the sequential order of single activities such as monitoring of parking lot scenarios [14] and human-object interactions in indoor settings [2][15]. In addition, Petri Nets (PNs) are presented to coordinate multiple activities and to model relations such as sequencing, and concurrency for outdoor video surveillance [16].

PGM approaches consider the uncertainty of the low-level analysis. Their structure is learned from training data or explicitly defined. Bayesian Networks (BNs) are PGM approaches assuming observable states that have been applied, among others, for person-person interactions [17] and outdoor surveillance [14]. However, BNs do not model temporal composition of events. Furthermore, Hidden Markov Models (HMMs) are proposed to combine the advantages of FSMs (temporal evolution) and BNs (probabilistic model) for non-observable states. Due to its simplicity and efficient parameter learning, they have been widely used for event recognition based on motion [18] and trajectories [19]. However, they are restricted to simple and sequential temporal patterns (Markovian model) and they may fail to recognize the same event performed in a different manner (as they rely on training data). Moreover, Dynamic Bayesian Networks (DBNs) have been introduced for temporal sequencing of BNs [20]. However, their use is limited due to the high computational complexity and the requirement of large amounts of training data.

2.3. Semantic Modeling approaches

SM approaches model an event as a structured description specified by the domain expert. These models are deterministic and the reasoning under uncertainty is not feasible. Among existing literature, Syntactic Models (SyM) represent complex events as hierarchical strings of symbols and detect them using simple routines such as Context-Free-Grammar (CFG) [21]. Moreover, Constraint Satisfaction Models (CSMs) define a set of rules derived from the hierarchical event description, among others, for airport monitoring [3], human-object interactions [22] and bank surveillance [23]. Another limitation of SM approaches is that they do not suggest the strategies to recognize the events from their descriptions.

2.4. Hybrid Modeling approaches

HM approaches combine the previously described categories for solving their limitations. For example, CFG extensions for handling low-level uncertainty have been proposed using BNs [7] and HMMs [24]. However, their descriptive capabilities are limited as they rely on the CFG approach and the employed analysis tools are domain specific (e.g., human interactions in close views that require specific training data [7]). No suggestion is given for their application to other domains or for the use of other low level tools. Furthermore, [6] extended the PN approach by measuring how well the event observations (e.g., PN states) fit to a predefined probability distribution of the observations for simple events. However, it assumes absolute certainty in the low-level analysis and does not define the computation of this distribution for complex events. Another enhancement of

Cat.	Approaches	Recognition from event descriptions	Uncertainty		Event		Real-time analysis	Domain specific
			Low-level	Semantics	Simple	Complex		
PR	KNN[12]	No	Yes	No	Yes	No	Yes	Yes
GM	BN [17]	No	Yes	No	Yes	No	No	Yes
	PN [16]	Partial	No	No	Yes	Yes	Yes	No
	DBN [20]	No	Yes	No	Yes	Yes	No	Yes
	FSM [15]	No	No	Yes	Yes	Partial	Yes	No
	HMM [18]	No	Yes	No	Yes	No	Yes	Yes
SM	SyM [21]	No	No	No	Yes	Yes	Yes	No
	CSM [3]	No	No	No	Yes	Yes	Yes	No
HM	P-PN [6]	No	No	Yes	Yes	Yes	-	No
	P-PN [5]	No	No	Yes	Yes	No	-	No
	CFG-BN [7]	No	Yes	Yes	Yes	Yes	No	Yes
	P-CSM[26]	No	No	Yes	Yes	Yes	-	No
	Proposed	Yes	Yes	Yes	Yes	Yes	Yes	No

Table 1: Comparison between the main reviewed approaches for video event recognition. (Key. PR: Pattern Recognition, GM: Graphical Modeling, SM: Semantic Modeling; HM: Hybrid Modeling, A: Automatic, M: Manual).

PNs is proposed in [5] for detecting events with variable duration. Simple semantics are defined without providing a structured model for semantic representation. Hence, their extension to describe complex events and consider low-level uncertainty is not straightforward. Probabilistic extensions are proposed to handle the uncertainty of simple event definitions for PNs [25] and CSMs [26]. In both approaches, the objective is to define a certainty measure for the observations associated to the event description components (e.g., map the certainty of a person being close to a zone of interest by using sigmoid [25] or Gaussian [26] functions). However, they do not consider the low-level uncertainty and they do not define the relation between the event descriptions and their recognition strategies.

Our approach fits into this category that combines semantic and probabilistic approaches. We propose a framework for representing and recognizing human-related video events. Its main contribution is a generic solution for event recognition in which event descriptions are converted into suitable recognition strategies that consider the uncertainty of low-level analysis. Unlike the reviewed literature, we formalize the principles to build graphical recognition models (generally ad-hoc requiring high level of expertise) from descriptions of simple and complex events. Therefore, we apply the most adequate strategy to each event type incorporating the uncertainty of low-level analysis. Furthermore, the uncertainty of the semantic definitions is addressed by including the specific recognition problems for each modeled event as proposed in [15]. Although our approach is demonstrated in the video monitoring domain, it is not restricted to a specific domain or implementation as opposed to many existing approaches. Table 1 summarizes and compares the main reviewed approaches.

Note that, although there exist other approaches for converting event descriptions into the PN formalism [16][27], our approach differs by building recognition strategies for short and long term human-related events as well as by providing a framework to consider the uncertainty of low-level analysis ([16] and [27] are only valid for long-term events assuming accurate analysis without considering its uncertainty). Hence, both approaches provide a partial solution for event recognition.

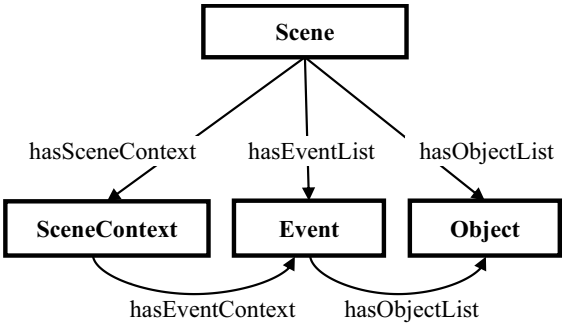


Fig. 1. Entity relationships exploited for event recognition.

3. Proposed framework

A complete framework has been designed for the event recognition task. In this section, we overview the event representation approach and the framework structure.

3.1. Event representation

For event representation, we have selected a state-of-art approach [9]. It is composed of an upper ontology that describes the structure of each knowledge type and leaves explicit the information that has to be inserted for modeling each domain. We use the *Scene* entity that represents the each domain by means of hierarchical descriptions of the scene objects (*Object* entity), their relations (*Event* entity) and additional information (*SceneContext* entity). We propose to exploit their relations (depicted in Fig. 1) for achieving an effective recognition of events.

The *Object* entity represents the physical scene objects. *Mobile* and *Contextual* objects are distinguished by their ability to initiate motion. Furthermore, *Contextual* objects are divided into *Fixed* and *Portable* objects (if they can be displaced). Therefore, events can be defined considering relations with moving entities (e.g., person), stationary objects (e.g., luggage) and fixed scene parts (e.g., open a window). The *Event* entity represents spatio-temporal relations between *Object* entities. Each *Event* entity is related to *Object* entities by the *hasObjectList* property. Furthermore, it is sub-classed depending on the number of agents involved (single and multiple) and the temporal relation with its events (simple and complex). In this work, we use the latter classification to efficiently organize the event recognition strategies. Hence, we develop strategies for recognizing simple and complex events. The *SceneContext* entity defines all the information that may influence the way a scene is perceived and can not be described using the *Object* and *Event* entities. In this work, we are interested in the *SpatialContext* and *EventContext* entities to provide, respectively, the scene layout and the event relations not described using the *Event* entity.

Among the related literature, the selected approach shares the basics for representing event-related semantics with the ViSOR ontology [28]. Both define entities such as objects, events and context. However, the ViSOR ontology is oriented to semantic annotation whereas [9] is focused on the description of the relations among entities. Hence, [9] defines events as spatio-temporal object interactions. Moreover, [9] defines different types of events (according to temporal and action thread characteristics) and contextual relations whilst ViSOR ontology only presents a list of concepts what prevents the use of the ViSOR ontology within the proposed approach. Finally,

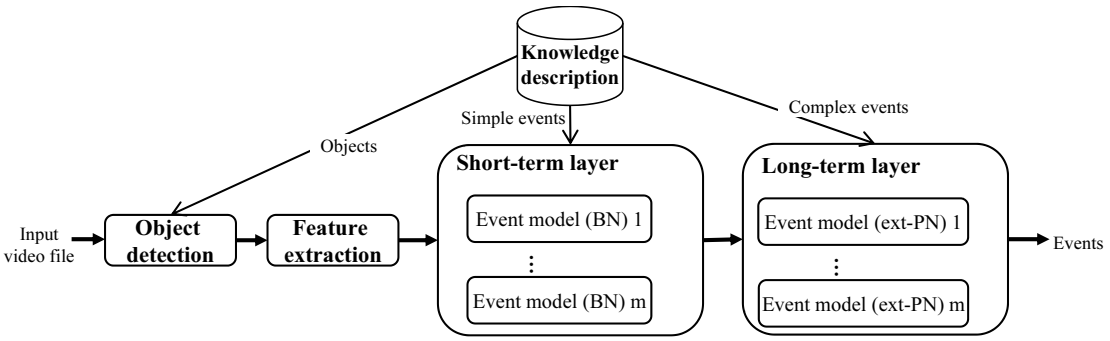


Fig. 2. Proposed framework for the recognition of events.

ViSOR ontology only describes domain knowledge and [9] includes additional knowledge sources (such as system capabilities) that allows to define the available recognition strategies.

3.2. Framework structure

In this paper we propose an event recognition framework composed of four modules as shown in the Fig. 2. The first module detects the objects of interest (i.e., the defined *Object* entities) from a video sequence. Then, the second module extracts the features required for event recognition. After that, a two-layer structure recognizes events considering the uncertainty of the analysis process being guided by the hierarchical event representation (described as in [9]). First, the *short-term* layer performs the detection of simple events that are characterized by their occurrence in short-time periods. A BN is defined for each event based on its description. Then, the *long-term* layer recognizes the complex events that present a temporal relation among its counterparts. A probabilistically extended PN is defined for each hierarchical event representation composed of simple and complex events. This event recognition structure is detailed in the section 4. The proposed combination addresses the limitations of the BN (not being able to model temporal event composition) and PN (deterministic detection) approaches. Note that this framework can fit the needs of a large variety of application domains by representing the prior knowledge and implementing the appropriate techniques for object detection and feature extraction.

4. Recognition of events

We propose a hybrid modeling to handle the low-level analysis uncertainty guided by the domain knowledge descriptions as defined in [9]. It establishes a common structure to recognize events that share similar characteristics. It consists of the *short-term* and the *long-term* layers that are described as follows.

4.1. Short-term layer

The *short-term* layer recognizes the events composed of hierarchical combinations of sub-events without temporal relations (e.g., *blob-inside-zone*). In the selected representation model [9], they correspond to the simple events types *SimpleWithSingleObject* and *SimpleWithMultipleObject*. This layer extends [8] by formalizing the building process of the recognition structure and its inference capabilities.

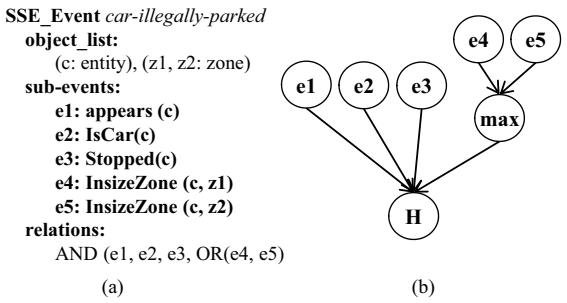


Fig. 3. Example for the *Car-illegally-parked* event. (a) Event description in which two zones were considered for detecting a car illegally parked. (b) Corresponding BN (bottom node and e_i nodes represent, respectively, the event occurrence and the described sub-events).

4.1.1. Layer modeling

For modeling this layer, we have chosen the BN approach as it offers several advantages such as the uncertainty handling and the knowledge-based structure definition. Each simple event is recognized with a hierarchical BN. The BN lower levels correspond to the observed features, the sub-events and their relations, whilst the upper level represents the event occurrences.

The BN structure is usually hard-coded relying on expert knowledge. This structure is represented with a directed acyclic graph (DAG), $\mathcal{G} = \langle \mathcal{N}, \mathcal{T} \rangle$, where $\mathcal{N}$ is the set of nodes representing the states and $\mathcal{T}$ is the set of transitions between states. We simplify this design by proposing a formal process based on the event description models [9], where each event is represented using relations between objects (sub-events) and events (spatial and logical).

For each event description, a root node is included to define its recognition with a binary value (denoted with H and $\tilde{H}$ for, respectively, indicate its occurrence or not). Then, additional nodes are included into the BN based on the event description (hereafter called evidences, E_i) as follows. Firstly, sub-events or feature changes that compose the event description are included in the network structure as nodes and are forced to produce an output probability $P(E_i/H)$ for indicating its contribution to the modeled event. This probability can be computed using a threshold function (e.g., the probability is either 0 or 1 if a feature value is above a threshold), other learned distribution forms (e.g., Gaussian or uniform) or using the likelihood of the associated classification problem (e.g., likelihood of the people recognition task). Secondly, spatial relations among the sub-events or the objects of the definition are included in the structure as additional nodes and their probability is computed using a threshold function. Finally, logical event relations are included in the BN structure. For each logical OR relation, an additional node is included connected to the root node. Then, the transitions of the nodes that compose this relation (e.g., sub-events) are redirected to this node (they were initially connected to the root node). Thus, the compounds of this logical relation are connected to the root node through the recently included node. Its probability is computed as the maximum value of all the incoming nodes $P(E_{OR}/H) = max(P(E_j/H))$. For each logical NOT relation, an additional node is placed between the negated sub-event and the root node. Then, its probability is computed as $P(E_{NOT}/H) = 1 - P(E_j/H)$. Finally, no operation is performed for the logical AND relation because it is intrinsically modeled in the BN approach. Algorithm 1 summarizes this building process for the *short-term* layer and Fig. 3 shows the description of the *Car-illegally-parked* event under the formalism proposed in [9] and the obtained BN structure.

Algorithm 1 Short-term layer structure composition

Input: Domain knowledge description D.
Output: A set of BNs, $S = \{BN_i\}$, that represent the short-term layer structure.

begin
$S = \{\emptyset\}$
for each simple event e_i in D **do**
  Add root node H_i to BN_i
  for each sub-event se_j of e_i **do**
    Add a node E_j in BN_i
    Connect the node E_j to H_i
  end for
  for each spatial relation sr_j of e_i **do**
    Add a node E_j in BN_i
    Connect the node E_j to H_i
  end for
  for each logical relation lr_j of e_i **do**
    Add a node E_j in BN_i
    Identify the E_k nodes that compose the relation
    Redirect the E_k node transitions to the node E_j.
    if logical relation is 'OR' **then**
      $P(E_j/H_i) = max(P(E_k/H_i))$.
    end if
    if logical relation is 'NOT' **then**
      $P(E_j/H_i) = 1 - P(E_k/H)$.
    end if
  end for
  $S = \{S,\ BN_i\}$
end for
end

4.1.2. Probability computation

The probability of a BN, composed of N variables $(H_1, ...H_N)$, is defined as a product of conditionally independent probabilities as follows:

$$P_{BN} = \prod_{i=1}^{N} P(H_i/pa(H_i)) \tag{1}$$

where $pa(H_i)$ is the set of parent nodes of H_i (i.e. those nodes directly connected to H_i via a single transition). This conditional probabilities can be learned from training data or derived from the process associated to each node. In this work, we compose a BN for each event and therefore, only one variable H_i exists in each BN (hereafter called H). The probability of the each BN corresponds to the computation of the $P(H/pa(H))$ term. In such case, $pa(H)$ represents the evidences $E_i, ...E_N$ of the event that are connected with the node H.

Finally, the BN probability, $P(H/E_{1...N})$, is computed using Bayesian inference as follows:

$$P_{BN} = P(H/E_{1...N}) = \frac{\prod_{i=1}^{N} P(E_i/H)P(H)}{\prod_{i=1}^{N} P(E_i/H)P(H) + \prod_{i=1}^{N} P(E_i/\tilde{H})P(\tilde{H})} \quad (2)$$

where H is the hypothesis (or event) and E_i are its linked evidences ($i = 1...N$). Similarly to [29], we assume no prior information about event occurrence ($P(H) = P(\tilde{H}) = 0.5$) and we use a default value for probabilities that are complex to estimate ($P(E_i/\tilde{H}) = 0.5$).

4.2. Long-term layer

The *long-term* layer recognizes events that span across frames and, therefore, they describe logical combinations of spatio-temporal relations. This layer complements the *short-term* layer by introducing temporal relations in the BN approach. In the selected representation model [9], they correspond to the *ComplexWithSingleObject* and *ComplexWithMultipleObject* events.

4.2.1. Layer modeling

For modeling this layer, we have selected the PN approach as it provides a robust formalism to express structured semantic knowledge. According to [16], it has several advantages such as its use for deterministic and stochastic inference of event occurrences, its top-down representation for the levels of abstraction of hierarchical semantic definitions (allowing sequencing, concurrency and synchronization) and its incremental event recognition without re-evaluating past event occurrences.

Similarly to the *short-term* layer, we reduce the high dependency on expert knowledge for the design of the PN structure by using descriptions of complex events. We recognize each complex event with a PN. Specifically, we use Plan-PNs [5] that model the occurrence of each sub-event (opposed to Object-PNs that represent the evolution of object features with a unique PN [5]). In a Plan-PN, the *places* represent sub-events and their occurrence is indicated by a *token* in a *place*. *Transition* nodes define the conditions for their recognition. Observe that the end of this sub-event recognition is not modeled in this approach (unless defined in the PN with a *transition*). Hence, we include the temporal information of the detected sub-events for each *token*. For modeling relations between sub-events, we employ *hierarchical* and *conditional transitions* [16]. The *Firing* status of these *transitions* indicates the recognition of the associated sub-events. Finally, a sink *place* is added for modeling the event occurrence. The structure of each PN is determined as follows.

Firstly, a root and a sink *places* are included in the PN structure to define the start and end of the recognition of the event. Secondly, the event description is inspected to identify its action threads, defined as a sequence of event executions performed by a single moving object (e.g., a car). Thirdly, each thread is processed to be included in the PN structure. For each one, two elements are included for the recognition of the moving object (e.g., car detection): a *conditional transition* and a *place*. This *transition* describes the classification problem (e.g., algorithm for car detection) and the *place* with its success (e.g., a car is detected). Then, the event relations of the thread (temporal, logical and spatial) are converted into the PN formalism. For the temporal relation *before* between two events, we use a chain of four sequential nodes (two *places* and two *transitions*). Starting from the previous *place* in the PN, we add a *conditional transition* for recognizing the first event and a *place* to indicate its occurrence. Then, we similarly include an additional *transition* and a *place* for the second event. For the other temporal relations (overlaps, during, starts, equals, meets and ends), we assume that an object can only perform one event at the same time and

Algorithm 2 Long-term layer structure composition

Input: Domain knowledge description D.
Output: A set of PNs, $L = \{PNe_i\}$, that represent the long-term layer structure.

begin
$L = \{\emptyset\}$
for each complex event e_i in D **do**
  Add a root *place* PR to PNe_i
  Add a sink *place* PS to PNe_i
  //Define each thread
  for each action thread a_j of e_i **do**
    Add a *transition* T_{j1} linked to a *place* P_{j1} to PNe_i for object recognition and its success
    for each temporal relation tr_k of a_j **do**
      Add sequences of four nodes
    end for
    for each spatial relation sr_k of a_j **do**
      Add a *transition* T_{jk} linked to a *place* P_{jk} to PNe_i
      Identify the P_n *places* of the related events
      Link the P_n *places* with the *transition* T_{jk}
    end for
    for each logical relation lr_k of a_j **do**
      Identify the P_n *places* of the related events
      if logical relation is 'AND' **then**
        Add a *hierch. transition* T_{jk} linked to a *place* P_{jk} to PNe_i
        Link the P_n *places* with the *transition* T_{jk}
      end if
      if logical relation is 'OR' **then**
        Add a *place* P_{jk} to PNe_i
        Link the P_n *places* with the *place* P_{jk}
      end if
      if logical relation is 'NOT' **then**
        Add a *transition* T_{jk} linked to a *place*P_{jk} to PNe_i
        Link the P_n *places* with the *transition* T_{jk}
      end if
    end for
  end for
  //Define thread relations
  for each temporal relation tr_k between a_j and a_i **do**
    Add a *transition* T_k for the temporal conditions to PNe_i
    Link the sub-event *places* to T_k
    Add a *place* P_k linked to T_k for its occurrence
  end for
  for each spatial relation sr_k between a_j and a_i **do**
    Develop the relation as for an action thread
  end for
  for each logical relation lr_k between a_j and a_i **do**
    Develop the relation as for an action thread
  end for
  $L = \{L, PNe_i\}$
end for
end

therefore, these relations define dependencies between different action threads. Their modeling is done by connecting the *places* of the two events to an additional *conditional transition* that checks the conditions over their temporal intervals as defined by Allen's Algebra [30] (this temporal data of each detected sub-event is available for each *token*). We also include a *place* to indicate the occurrence of the relation. For spatial relations, a *transition* and a *place* are included in the PN to represent, respectively, this relation and its occurrence. Then, arcs are drawn from the *places* of the events that compose the relation to this additional *transition*. Logical relations are straightforward to model using the PN formalism. The logical AND relation is modeled as incoming arcs (from the related events) connected to an included *hierarchical transition*. The logical relation OR is modeled as a *place* with incoming arcs from the *transitions* corresponding to the events of the relation. The logical NOT relation is defined as a condition included in the corresponding *transition* and the event probability is modified with its complementary value similarly to the *short-term* layer. Finally, the junctions between action threads are established as defined in the event description. For converting these thread relations, we use the previously mentioned rules defined for the event relations. Algorithm 2 summarizes this composition procedure.

In addition, we overcome well-known event recognition problems or uncertain event definitions by including solutions in its definition as suggested by [15]. However, this operation is hard to be formalized as it relies on expert knowledge. Sub-section 5.2 illustrates an example of this strategy in which a PN represents the *Abandoned-object* event.

Fig. 4 shows the description and the corresponding PN of the *Pickup-train* event. First, three *transitions* are included on the top of the PN to describe the classification stage for each object involved. Then, sub-events are represented as *conditional transitions* and connected to *places* through *arcs*. These connections are guided by the relations given in the event definition. As it can be observed, the temporal relation *before* is represented using a sequential combination of *places* and *transitions*. The OR logical relation is represented as incoming *arcs* from two *transitions* (T8 and T9) to the P6 *place*. The recognition of the event is defined by the AND operator that is represented with two incoming *arcs* from *places* P6 and P7 to *transition* T11. Observe that *transition* T11 is marked with a *null* as it is a *hierarchical transition* (it fires when all of its input *places* have at least one *token*).

4.2.2. Probability computation

Standard PNs do not handle the uncertainty associated to the analysis. As a first approach, we propose a simple combination of probabilities to include the uncertainty of the event recognition. We assume that a probability $P(T_i)$ is obtained from each activated *transition* T_i. This probability can come from a simple event (modeled as a BN), a complex event (modeled as a PN) or the relations between them (temporal, logical and spatial). For logical relations, we compute their probability using the following rules:

$$P(T_i) = \begin{cases} \prod\limits_k P(T_k) & if \quad T_i \Longleftrightarrow AND \\ \max\limits_k (P(T_k)) & if \quad T_i \Longleftrightarrow OR \\ 1 - \frac{1}{k}\sum\limits_k P(T_k) & if \quad T_i \Longleftrightarrow NOT \end{cases} \tag{3}$$

where T_i is the *transition* introduced for the logical relation, $P(T_i)$ is the resulting probability and $P(T_{1...k})$ are the probabilities of the k *transitions* connected to T_i (through *arcs* and *places*).

Traditional *transition* activation (or *firing*) is deterministically performed (i.e., the associated

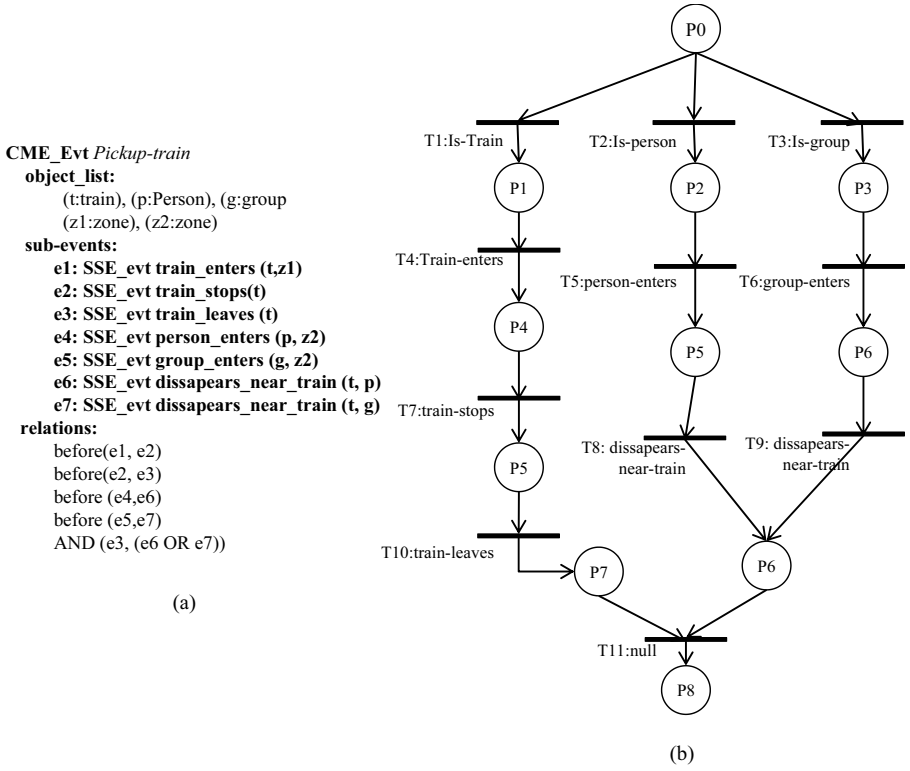


Fig. 4. (a) Description of the *Pickup-train* event (complex and multiple thread). (b) Corresponding Petri Net (each PN conditional transition corresponds to each sub-event of the description). Note that the transitions T1, T2 and T3 have been included in the top of the PN to define the classification of each object involved in the event. Additionally, a hierarchical transition T11 is included at the bottom of the PN to reflect the recognition of the event.

conditions are satisfied). For considering low-level uncertainty, we use a confidence level to threshold $P(T_i)$ as follows:

$$Firing_i = \begin{cases} 1 & if \quad P(T_i) > \tau_i \\ 0 & if \quad P(T_i) \leq \tau_i \end{cases} \tag{4}$$

where T_i is the *transition* to be *fired* and τ_i is a threshold (confidence value). This operation allows to reduce the computational load of the event recognition structure by discarding events with low probability that can be due to errors of the low-level analysis (e.g., non-accurate object extraction). Note that a specific threshold can be defined for each PN *transition* based on expert knowledge (e.g., determining the error-prone event recognition modules). In this work, we use the same value for all the *transitions* in the PNs ($\tau_i = 0.1$).

Finally, event probability is obtained when a *token* reaches the PN sink *place* as follows:

$$P(H/T_{1...N}) = \frac{1}{N}\sum_{i=1}^{N} P(T_i) \tag{5}$$

where $P(H/T_{1...N})$ is the probability of the event H, $T_{1...N}$ are the *fired transitions* that the *token* has passed, N is the number of *fired transitions* and $P(T_i)$ is their probability.

4.3. Contextual information

Furthermore, we extend the ontology by implementing the *SceneContext* entity that defines all the information that affects the way a scene is analyzed and consists of the *SpatialContext, ObjectContext* and *EventContext* concepts. In our work, we use the *SpatialContext* concept to provide the initial environment layout in terms of the existing *Contextual Objects* and to define the location of events (e.g., leave objects on the *Table* object). *ObjectContext* concept determines relations between objects for each specific scenario (e.g., the size ratio between *Mobile* and *Contextual* objects). *EventContext* concept is used to define relations between events (e.g., mutually exclusive events or predefined occurrence order for events). These context definitions are introduced to save computational cost (i.e., not analyzing events that can not happen due to the model constraints) and decrease the false positive event rate by limiting the system response (i.e., adding more conditions for event occurrence).

5. Application to the video monitoring domain

We demonstrate the proposed framework for recognizing human-object interactions in the video monitoring domain. In this section, we overview the selected tools for object detection and feature extraction as well as the defined events.

5.1. Object detection and feature extraction

Currently, the processing capabilities rely on the analysis proposed by [8]. Firstly, it applies background subtraction and then, shadows are removed from the foreground segmentation map. After blob extraction, a first-order Kalman filter is used for blob tracking. Finally, several blob-based features are extracted to feed the proposed event recognition framework. Further details about the extracted features are given in [8].

5.2. Event modeling

For this domain, we have modeled three simple and two complex human-object interactions.

The three simple events are *Leaves-object*, *Gets-object* and *Uses-object*. Their occurrence is determined for each frame. For their definition, we have specified some simple routines that compose their representation. Thus, the *BelongToFG* routine uses the feature *Foregroundness* to indicate the degree of belonging to foreground by means of a trained Gaussian model. In a similar way, the *BelongToBG* routine uses the feature *Backgroundness* to provide a probability of belonging to the background. Moreover, *IsOwner* routine calculates the agent (e.g., person) that is performing the event and interacting with an object (e.g., blob with low *PeopleLikelihood*). This owner is detected as the closest blob with high-people likelihood and determined when the object appeared in the scene. The *IsPerson* routine uses the *PeopleLikelihood* feature. The *IsContextualObj* routine checks if the blob under analysis belongs to the defined *Contextual Object* entities by using the *Compactness* feature and specific appearance-based models (if the information is available). The *OverlapSkinRegion* routine calculates the spatial overlap between an entity and the skin regions of another entity. Finally, the *Stopped* routine uses the *BlobVelocity* feature to determine whether a blob is moving or not. Fig. 5 presents their descriptions and the associated (naïve) BNs. As it can

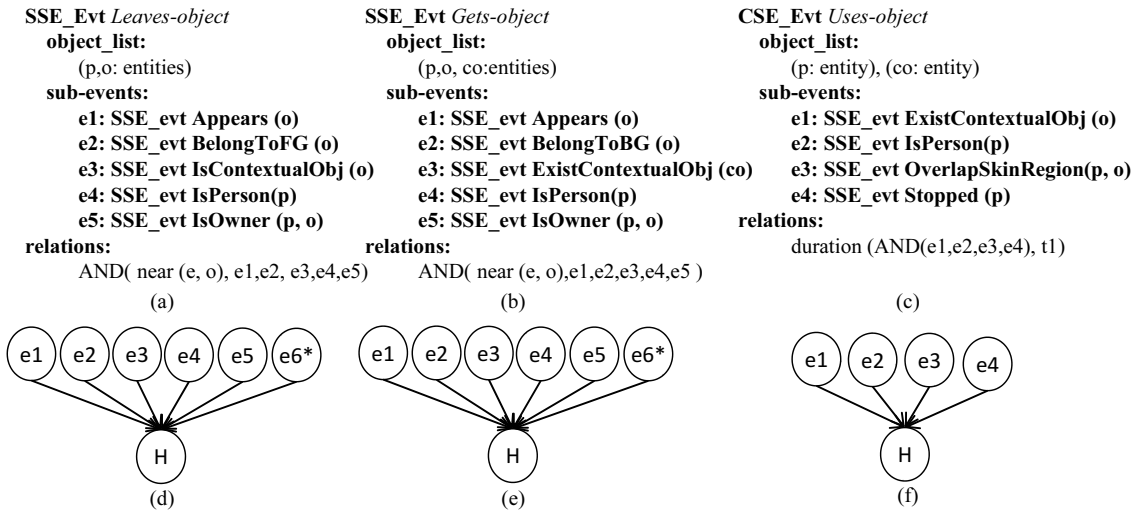


Fig. 5. Events models for the experiments in controlled environments. Data correspond to the semantic definition of the (a) *Leaves-object*, (b) *Gets-object* and (c) *Uses-object* events. Their naïve BN are depicted in, respectively, subfigures (d), (e) and (f). Nodes marked with * represent the spatial relation *near*.

be observed, additional nodes are included in the BN to represent the spatial relation *near* (marked with *). Furthermore, the *Uses-object* event included the relation *duration* that defines the length of event using the parameter *t1* to detect its occurrence.

For complex events, two common events in public video surveillance have been described: *Abandoned-object* and *Stolen-object*. Fig. 6 depicts the description and the PN for the *Abandoned-object* event. As proposed by [15], we overcome existing event recognition problems by including strategies for their solving. For the *Abandoned-object* PN, the left side defines the typical model for detecting abandoned objects [31] whilst the right side describes their detection considering that the action owner is not likely to be identified (difficult in crowded scenarios). In addition, PN loops correspond to two temporal relations: *before* (*before(e4, e5)* and *before(NOT(e5), e4)*). Besides, the *stationary* routine is defined to detect stationary objects for a given time period, represented by the relation *duration(e9, 30)* (i.e., remains stationary for 30 seconds) and uses the *BlobStationarity* feature. The *Stolen-object* event definition and its associated PN are similarly defined by replacing the *Leaves-object* and *BelongToFG* events with the *Gets-object* and *BelongToBG* events.

5.3. Contextual information

For providing such information, we assume that two kind of environments exist in the video monitoring domain: controlled and uncontrolled. The former consists on the monitoring of places characterized by the presence of few people and the availability of contextual information useful for event recognition such as the object types that can appear. The latter covers the monitoring of public places that are usually crowded (e.g., train stations). They present high data variability (as opposed to the meeting domain) and few contextual information is available. In this sub-section, we describe the contextual information defined for each situation.

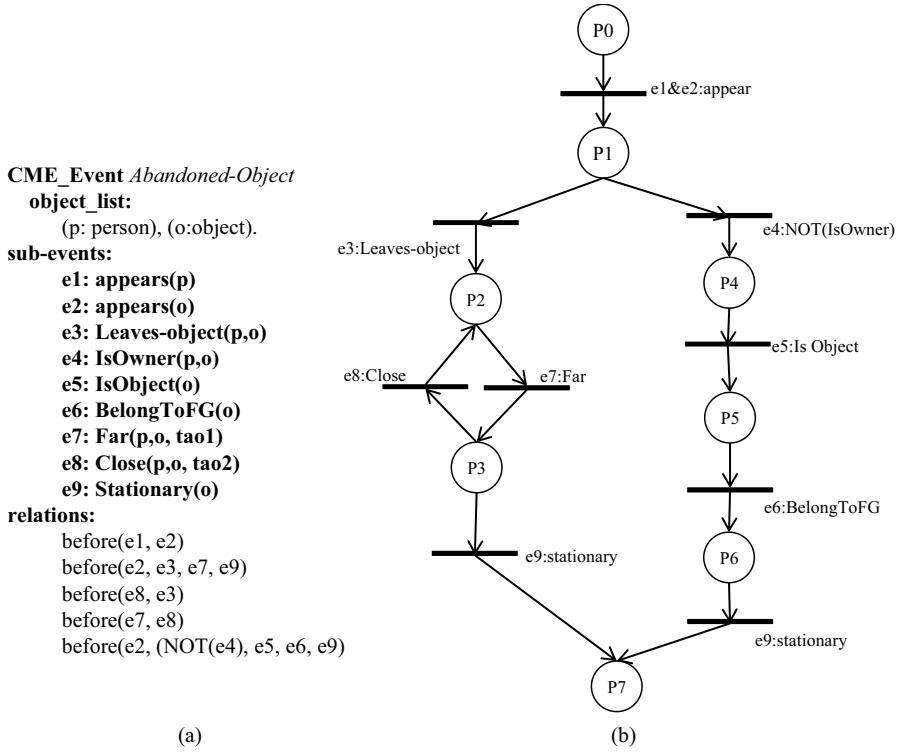
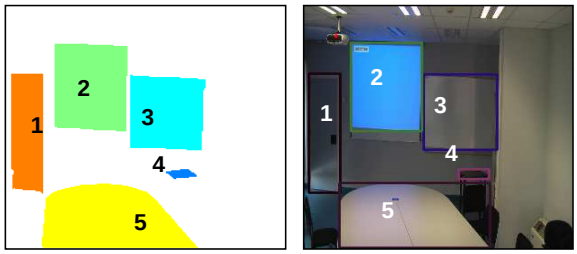


Fig. 6. Complex event *Abandoned-object* modeled for the video surveillance domain. Data correspond to (a) semantic definition and (b) the corresponding PN.

5.3.1. Controlled environments

For the *Object* entity, we distinguish *Mobile Objects* (*Group* and *Person*) and *Contextual Objects*. Among the latter, we discriminate between *Fixed* (*Wall, Window, Floor, Table*, *Door, Blackboard, Screen* and *ProjectionArea*) and *Portable Objects* (*Chair*, *Laptop*, *MobilePhone* and *Generic*).

Furthermore, the *SceneContext* entity is exploited to represent the contextual information of this scenario. *SpatialContext* is used to provide the location of the existing *Contextual Objects*. An example of such layout is depicted in Fig. 7. Moreover, the *ObjectContext* entity determines relations between objects for each specific scenario (e.g., the size ratio between *Mobile* and *Contextual* objects for detecting *Groups*). The *EventContext* entity is used to define relations between events (e.g., mutually exclusive events or predefined occurrence order for events). Currently, we have implemented a constraint for the event location by using the spatial relation *overlap* between the *Leaves-object* event and the *Table* object. Furthermore, the *ExistContextualObj* routine, that checks if a *Contextual* object exists in the same spatial location as the blob under analysis, is included as contextual information for the *Gets-object* and *Uses-object* events. These constraints are included in the BN of each event as additional nodes. Observe that these constraints require the knowledge of the location of the existing objects and the fixed elements of the scene (that is impractical for uncontrolled environments). They are introduced to decrease the false positive event rate by adding constraints to recognize the event and to save computational cost by avoiding possible occurrences



Object tags **Bounding Boxes**

Object categories

1 Object\ContextualObject\FixedObject\Door
2 Object\ContextualObject\FixedObject\ProjectionA
3 Object\ContextualObject\FixedObject\Blackboard
4 Object\ContextualObject\FixedObject\Table
5 Object\ContextualObject\FixedObject\Table

Fig. 7. Annotated scene layout example.

of complex events that include the context-modified simple events (e.g., *Abandoned-object* event includes the *Leaves-object* event).

5.3.2. Uncontrolled environments

Unlike the controlled situation, little prior information is available. Although it is assumed that some types of objects are known (e.g., person, trains), the variability of the features of the objects of interest is not known (e.g., luggage appearance). Moreover, the relation of events with the layout of the scene is more difficult to estimate as, in these settings, the recording device (e.g. camera) is typically placed at a medium or long distance from the action. Therefore, the type of events that can be recognized with enough accuracy is limited to the ones related to trajectory analysis in most of the cases. For these reasons, objects and events have to be defined in general terms without considering detailed types. Due to complexity of extracting contextual information in this situation, the *SceneContext* entity is not exploited.

For the *Object* entity, we distinguish *Mobile Objects* (*Group* and *Person*) and *Contextual Objects*. Among the latter, we discriminate between *Fixed* (*Wall, Window, Floor,* and *Area*) and *Portable Objects* (*generic*).

6. Experimental results

We evaluate our approach on the video monitoring domain for the controlled and uncontrolled conditions modeled in the previous section. It has been implemented using the OpenCV library[1]. Tests were performed on a standard PC (P-IV 2.8GHz and 2GB RAM).

For comparison purposes, we have selected the widely used CSM approach [3][23][31][22][32]. It defines rules for each component of the event description that provide a binary decision on whether these components happened or not. These rules are based on thresholding the confidence of the

[1] `http://sourceforge.net/projects/opencvlibrary/`

analysis task or feature associated to the event description component. However, the evaluation of the accuracy of the event recognition strategies requires to use the same low-level analysis to be independent to the different analysis (or features) proposed in each work. In our case, we have applied the same blob-based analysis for object and feature extraction. Then, we have implemented two CSM approaches for the conditions considered in the experiments.

6.1. Performance evaluation criteria

For matching event annotations and detections, we have used a criteria defined as follows:

$$Match(E^{GT},\, E^{D}) = \begin{cases} 1 \quad if & score > \rho & \bigwedge \\ & \left|T^{D}_{start} - T^{GT}_{start}\right| < \tau_1 & \bigwedge \\ & \left|T^{D}_{end} - T^{GT}_{end}\right| < \tau_2 & \bigwedge \\ & \frac{2\left|A^{GT} \cap A^{D}\right|}{|A^{GT}|+|A^{D}|} > \sigma & \\ 0 & Otherwise & \end{cases} \tag{6}$$

where E^{GT} and E^{D} are the annotated and detected events; *score* is the probability of the detected event; $(T^{D}_{start}, T^{D}_{end})$ and $(T^{GT}_{start}, T^{GT}_{end})$ are the frame intervals of the annotated (GT) and detected (D) events; $\left|A^{GT}\right|$ and $\left|A^{D}\right|$ represent the average area (in pixels) of each event; $\left|A^{GT} \cap A^{D}\right|$ is their average spatial overlap (in pixels); ρ, τ_1, τ_2 and σ are positive thresholds (heuristically set to the values $\rho = 0.75$, $\tau_1 = \tau_2 = 100$ and $\sigma = 0.5$). For event annotation and performance evaluation, we have used the ViPER toolkit [33].

For evaluation purposes, we use the Precision (P) and Recall (R) measures. Precision is the ratio between the correct and the total number of detections. Recall is the ratio between the correct detections and the total number of annotations. We also use the F-score measure, β, to combine Precision and Recall which is defined as follows:

$$\beta = 2 \cdot \frac{P \cdot R}{P + R} \tag{7}$$

6.2. Controlled environments

For this situation, we evaluate the proposed approach for the recognition of the previously defined three simple events: *Leaves-object* (LEA), *Gets-object* (GET) and *Uses-object* (USE). For comparison purposes, we have selected [22] (hereafter called CSM1) that defines two human-object interactions, insertion and removal, that are similar to, respectively, the *Leaves-object* and *Gets-object* events. Their detection is based on three simple rules: the detection of a blob splitting into two blobs, the detection of a static blob (i.e., same position for 10 frames) and the computation of the edges around the boundaries of the static blob to decide whether it is an insertion or a removal.

6.2.1. Dataset

A dataset has been collected from selected sequences of the VISOR[2], the HERMES[3], the WCAM[4], the CANDELA[5] and the MR[6] public datasets. For discussing the achieved results,

[2] `http://www.openvisor.org/`
[3] `http://iselab.cvc.uab.es/indoor-cams`
[4] `http://wcam.epfl.ch/`
[5] `http://www.multitel.be/~va/candela/abandon.html`
[6] `http://www-vpu.eps.uam.es/EDds/`

Fig. 8. Sample frames of the selected content for the experiments in controlled environments. Rows 1, 2 and 3 correspond to categories C1, C2 and C3. (From top-left to bottom-right) *AbandonedObject* (VISOR), *Indoor_activity1* (WCAM), *S1_0001* (MR), *Indoor_activity2* (WCAM), Cam1_indoor (HERMES), *S2_0004* (MR), *S3_0001* (MR), *Indoor_1.07* (CANDELA) and *S3_0002* (MR).

we have classified the sequences into three categories attending to an initial complexity estimation of the analysis stages that compose the proposed framework. Table 2 summarizes all the selected content (in terms of number of frames, annotated events and estimated complexity) and Fig. 8 show sample frames of selected sequences.

6.2.2. Results

In total, our approach detected 657 event occurrences. Fig. 9 shows their probability distribution. As it can be observed, a high amount of events are detected with extremely low probability ($score < 0.1$). They can be easily discarded as most of them correspond to false detections. However, the events with intermediate probability ($0.2 < score < 0.8$) present high uncertainty as it is difficult to decide whether they are correct or not. Low event probability can be due to non-accurate

Cat.	Frames	Events Occurrence			Complexity			
		LEA	GET	USE	FG	BT	FE	ED
C1	41753	28	21	9	M	L	M	M
C2	36446	25	15	10	M	M	M	H
C3	35570	29	28	35	V	H	V	V
Total	113769	82	64	54	-	-	-	-

Table 2: Dataset description for controlled environments (Key: LEA: *Leaves-object*, GET: *Gets-object*, USE: *Uses-object*, FG:Foreground seg., BT:Blob tracking, FE:Feature Extraction, ED:Event recog., L:Low, M:Medium, H:High and V:Very High).

Cat.	LEA						GET						USE					
	CSM1			Proposed			CSM1			Proposed			CSM1			Proposed		
	P	R	β	P	R	β	P	R	β	P	R	β	P	R	β	P	R	β
C1	.96	.85	.90	.93	.93	.93	.95	.90	.92	.90	.95	.92	-	-	-	1	1	1
C2	.77	.68	.72	.74	.80	.77	.72	.53	.61	.71	.66	.68	-	-	-	.63	.70	.64
C3	.40	.34	.36	.48	.51	.49	.50	.21	.29	.58	.35	.43	-	-	-	.58	.40	.47
Total	.74	.65	.69	.71	.74	.72	.76	.51	.61	.75	.63	.68	-	-	-	.68	.55	.37

Table 3: Recognition results for the analysis of controlled environments (Key: LEA: *Leave-object*, GET: *Get-object*, USE: *Use-object*, CSM1: rule-based approach [22]).

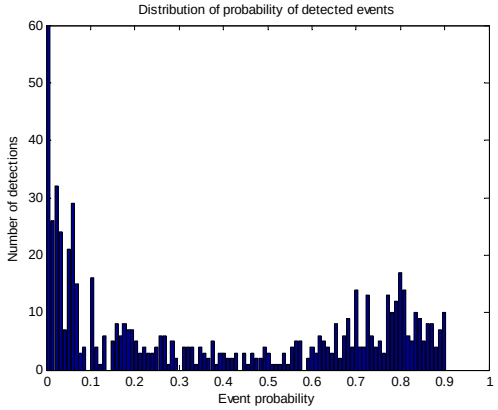


Fig. 9. Probability distribution of the detected events for the experiments in controlled environments. In total, 657 events were detected (filtered to 183 with a threshold of $\rho = 0.75$).

low-level analysis or event occurrences that can not be described with their semantic models (i.e., event model inconsistencies). Finally, we filtered the initial detections by thresholding the event probability with $\rho = 0.75$ (see Eq. 6), obtaining 183 event occurrences.

The obtained results are reported in Table 3. The framework presents high figures for scenarios in which foreground blobs are well detected and tracked; so the events can be easily recognized (C1 and C2). Furthermore, the included spatial constraints (*SceneContext* entity) increases the accuracy of the results by avoiding false detecting in non-predefined locations. Figures notably decrease in complex scenarios (C3) mainly due to multiple occlusions, group blobs and segmentation

Cat.	FG	BT	FE	ED	Total
C1	26.2 (59.4%)	0.5 (1.1%)	16.2 (36.7%)	1.2 (2.7%)	44.1 (100%)
C2	25.4 (47.8%)	0.9 (1.6%)	24.4 (45.9%)	2.4 (4.5%)	53.1 (100%)
C3	25.8 (42.6%)	2.1 (3.4%)	29.1 (48.1%)	3.5 (5.7%)	60.5 (100%)

Table 4: Average system execution time for controlled environments (ms) (Key: FG:Foreground segmentation, BT:Blob tracking, FE:Feature Extraction and ED:Event detection).

Cat.	Frames	Occurrences		Complexity			
		ABA	STO	FG	BT	FE	ED
C1	204408	54	52	L	L	M	M
C2	43632	10	4	M	M	M	M
C3	40951	15	-	H	H	H	M
C4	61951	14	-	V	V	V	V
Total	350942	93	56	-	-	-	-

Table 5: Dataset description for uncontrolled environments (Key: ABA:*Abandoned-object*, STO:*Stolen-object*, FG:Foreground seg., BT:Blob tracking, FE:Feature Extraction, ED:Event recog., L:Low, M:Medium, H:High and V:Very High).

errors (resulting in the fragmentation of foreground blobs). Additionally, non-modeled events (e.g., sitting or standing) adversely affect the detection of the modeled events. Compared to CSM1, we can observe that our approach improves the overall performance of the event recognition (β measure). However, CSM1 obtains higher Precision in simple categories due to the defined hard rules. This enhancement is reduced with increasing scenario complexity and for the C3 category, the proposed approach gets better accuracy. Sample results are shown in Fig. 10.

The computational cost of the proposed approach is summarized in Table 4; data correspond to the average execution time for each category and stage (normalized to the size of 320x240). As it can be seen, real-time analysis is achieved with an execution time between 44.1 ms (22.6 fps) and 60.5 ms (16.6 fps) for the best and the worst cases (categories C1 and C3 respectively). The foreground segmentation stage, FG, has a (quasi) constant computational cost independently on the sequence complexity because it works at pixel-level and is blob-independent. On the contrary, blob-level analysis, BT to ED, presents a dependency on the quantity and size of blobs of interest in each sequence being feature extraction (FE) the most execution time demanding stage.

6.3. Uncontrolled environments

For this situation, we evaluate the accuracy of the proposed approach for the recognition of the previously defined two complex events: *Abandoned-object* (ABA) and *Stolen-object* (STO). For comparison purposes, we have selected [32] (hereafter called CSM2) that defines a rule-based detection of the two complex events. It uses the following rules: the detection of a static blob (i.e., same position for 30 frames), the detection of the blob as non-people (by means of a specific classifier), the detection of a person as the individual of the action and the use of edge and color information of the static blob to decide whether it is abandoned or stolen.

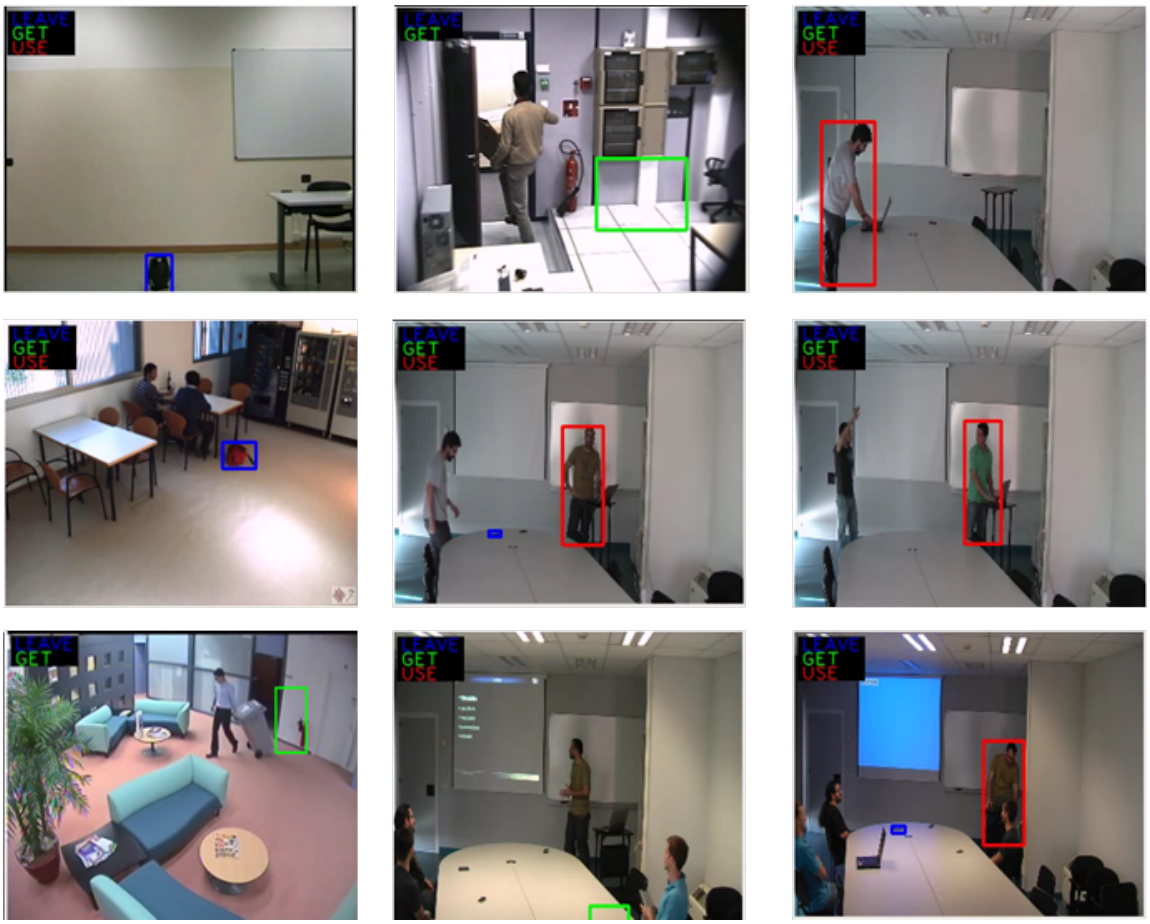


Fig. 10. Event detection examples for controlled environments. Rows 1, 2 and 3 correspond to categories C1, C2 and C3. (From top-left to bottom-right): *VISOR_AbandonedObject_06* (frame 213), *WCAM_indoor_activity_3* (frame 1121), *S1_0003* (frame 1974), *HERMES_cam1_indoor* (frame 616), *S2_0004* (frame 2582), *S2_0006* (frame 2737), *CANDELA_1.04* (frame 260), *S3_0001* (frame 7263) and S3_0002 (frame 5790). The color codes correspond to the *leave-object* (blue), *get-object* (green) and *use-object* (red).

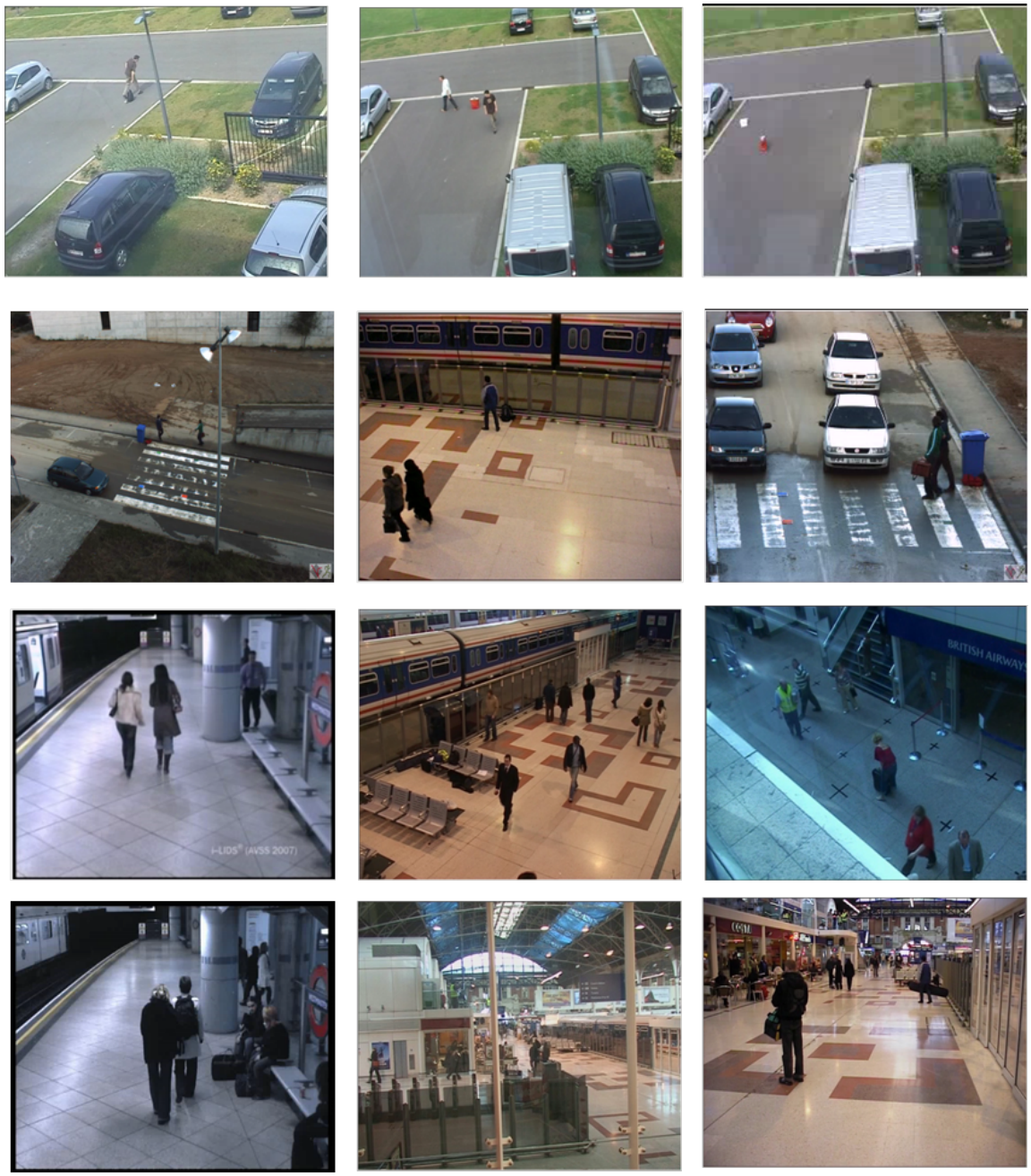


Fig. 11. Sample frames of the selected content for the experiments in uncontrolled environments. Rows 1, 2, 3 and 4 correspond to categories C1, C2 C3 and C4. (From top-left to bottom-right) *CantataMultitelCam1_018* (CANTATA), *CantataMultitelCam2_004* (CANTATA), *CantataMultitelCam2_016* (CANTATA) , *Cam2_outdoor* (HERMES), *S2-T3-C_3* (PETS2006), *Cam5_outdoor* (HERMES), *AVSS_AB_Easy* (AVSS2007), *S2-T3-C_4* (PETS2006), *S7_abandoned_bag* (PETS2007), *AVSS_AB_EVAL* (AVSS2007), *S2-T3-C_1* (PETS2006), and *S2-T3-C_2* (PETS2006).

6.3.1. Dataset

The evaluation dataset is composed of sequences from the CANTATA[7], HERMES[8], i-LIDS for AVSS2007[9], PETS2006[10] and PETS2007[11] public datasets. These sequences range from simple sequences with one individual to challenging sequences in crowded situations. For solving the well-known problem of background initialization [34], each sequence was preprocessed using a median filter to capture this background. Additionally, a region of interest was defined for each sequence to indicate the possible location of the event[12] (contextual information).

Similarly to the previous experiment, we have classified the sequences into four categories attending to an initial complexity estimation of each analysis stage of the proposed framework. Table 5 summarizes all the selected content (in terms of number of frames, annotated events and estimated complexity) and Fig. 11 shows sample frames of selected sequences.

6.3.2. Results

In total, our approach detected 2229 event occurrences. This high number of detections can be explained by the absence of the context-based constraints (opposed to controlled environments). Fig. 12 shows their probability distribution. As it can be observed, the probability of the events are concentrated in two ranges of values. The first one consists of events with low probability ($score < 0.1$) and they can be easily discarded as most of them are due to small segmentation errors. The second concentration is observed for intermediate-high probability ($0.6 < score < 0.8$). During experiments, it was observed that some of them were correct and some of them were due to wrong analysis of the classification modules (e.g., people recognition). Additionally, 274 events fell into an intermediate-low value ($0.2 < score < 0.6$) presenting a high uncertainty and therefore, additional mechanisms should be used for accepting or rejecting them. After filtering their probability by using a value of $\rho = 0.75$ (see Eq. 6), 202 event detections were considered as valid.

The obtained results are summarized in Table 6. It shows how event recognition in simple situations, such as category C1 and C2, performed reasonably well. On the contrary, the performance decreased in complex situations such as crowded scenarios. The high number of objects (moving and stationary) and the occlusions between them are the main problems that affect all the segmentation and tracking of moving objects. A high number of *False Positives* is obtained and the *Precision* measure is decreased. A post-processing stage would be desirable to filter these detections. However, the system is able to recognize most of the events presenting acceptable *Recall* values for complex categories. Compared to CSM2, our approach obtained better results demonstrating that the rules applied are less robust to recognize the event under the uncertainty of the low-level analysis. Hence, CSM2 accuracy highly decreases as the complexity of the scenario increases (and the low-level uncertainty). These results demonstrate that the modeling of relations Fig. 13 shows event recognition examples for the different categories. It should be noted the increase in the number of false positives as we analyze more complex categories.

The computational cost of the proposed approach is summarized in Table 7; data correspond to the average execution time for each category and stage (normalized to the size of 320x240).

[7] `http://www.multitel.be/~va/cantata/LeftObject/`

[8] `http://iselab.cvc.uab.es/indoor-cams`

[9] `http://www.avss2007.org/`

[10] `http://www.cvg.rdg.ac.uk/PETS2006/`

[11] `http://www.cvg.rdg.ac.uk/PETS2007/`

[12] This was mainly done to avoid detections in highly reflective surfaces or non-interesting spatial locations.

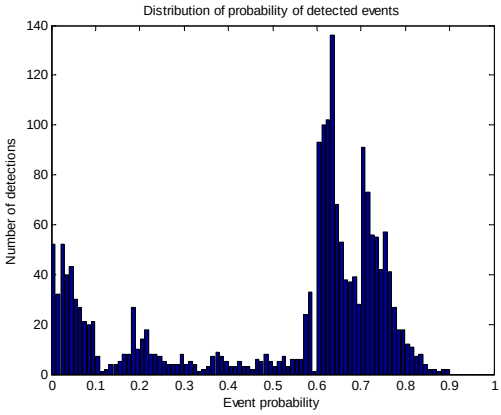


Fig. 12. Probability distribution of the detected events for the experiments in uncontrolled environments. In total, 2259 events were detected (filtered to 202 with a threshold of $\rho = 0.75$).

Cat.	**ABA**						**STO**					
	CSM2			**Proposed**			**CSM2**			**Proposed**		
	P	**R**	β	**P**	**R**	β	**P**	**R**	β	**P**	**R**	β
C1	.85	.90	.87	.85	.94	.89	.91	.80	.85	.85	.86	.85
C2	.44	.40	.41	.57	.80	.66	.28	.50	.35	.33	.75	.45
C3	.21	.46	.28	.24	.67	.35	-	-	-	-	-	-
C4	.16	.21	.18	.23	.42	.29	-	-	-	-	-	-
Total	.53	.67	.59	.52	.79	.62	.83	.78	.80	.77	.86	.82

Table 6: Recognition results for the analysis of controlled environments (Key: ABA: *Abandoned-object*, STO:*Stolen-object*, CSM2: rule-based approach [32]).

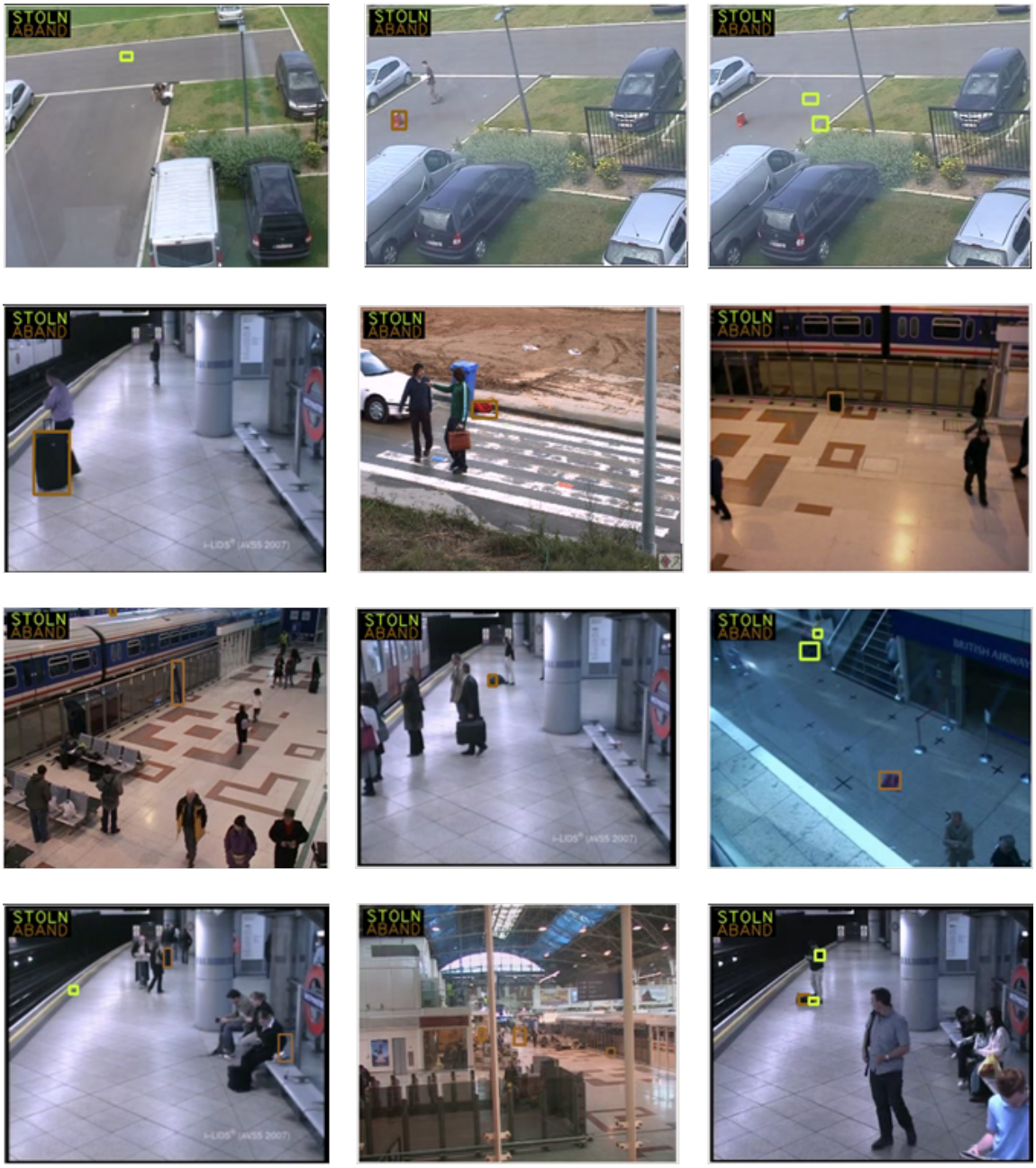


Fig. 13. Event detection examples for uncontrolled environments. Rows 1, 2, 3 and 4 correspond to categories C1, C2, C3 and C4. (From top-left to bottom-right): *CantataMultitelCam2_018* (frame 950), *CantataMultitelCam1_013* (frame 1548), *CantataMultitelCam1_013* (frame 1745), *AVSS_AB_Easy* (frame 2451), *HERMES_Cam3_outdoor* (frame 972), *PETS06_S7_T6_B3* (frame 1641), *PETS06_S5_T1_A4* (frame 2128), *AVSS_AB_Medium* (frame 2332), *PETS07_S7* (frame 1755), *AVSS07_hard* (frame 3543), *PETS06_S6_T3_H3* (frame 2329) and *AVSS_AB_EVAL* (frame 13430). The color codes correspond to the *Abandoned-object* (brown) and *Stolen-object* (yellow).

Cat.	FG	BT	FE	ED	Total
C1	26.0 (59.4%)	0.4 (1.1%)	16.4 (37.4%)	1.0 (2.2%)	43.8 (100%)
C2	25.9 (43.6%)	1.1 (1.8%)	28.9 (48.6%)	3.5 (5.8%)	59.4 (100%)
C3	25.8 (39.4%)	2.4 (3.6%)	32.2 (49.2%)	4.8 (7.3%)	65.4 (100%)
C4	26.1 (29.4%)	6.5 (7.3%)	48.4 (54.6%)	7.4 (8.3%)	88.5 (100%)

Table 7: Average system execution time for uncontrolled environments (ms) (Key: FG:Foreground segmentation, BT:Blob tracking, FE:Feature Extraction and ED:Event detection).

As it can be seen, real-time analysis is achieved with an execution time between 43.8 ms (22.8 fps) and 88.5 ms (11.3 fps) for the best and the worst cases (categories C1 and C4 respectively). Similarly to controlled environments, pixel-based analysis stages such as foreground segmentation present a (quasi) constant computational cost. The rest of the stages are blob-based and therefore, their computational cost varies with the complexity of the sequence (e.g., the number of blobs). A notable increase of the computational cost can be observed as compared with the controlled situation (between 11-46%) due to the high density of moving objects in the scene.

7. Conclusion

This paper has described a single-view video event recognition framework guided by hierarchical event descriptions. It has been presented how the formalization of knowledge relevant to video analysis within a specific domain can be used to define strategies for the event recognition. A two-layer strategy is proposed to recognize events handling the uncertainty of the low-level analysis. The *short-term* layer uses hierarchical BNs to recognize timeless events that consist of changes in object features. The *long-term* layer is in charge of detecting events with a temporal relation among their counterparts by using the PN approach. A simple extension of the basic PN structure is proposed to manage uncertainty obtained by the sub-events (related with the uncertainty of the low-level analysis). Formalisms are proposed to obtain the graphical recognition models (BNs and PNs) from event descriptions.

The accuracy of the proposed framework has been tested for the recognition of human-object interactions in controlled (short-term events) and uncontrolled environments (long-term events) for the video monitoring domain. The results showed that the proposed approach outperformed the traditional rule-based approach. A high recognition rate was achieved by exploiting the spatial relations between the persons and the scene layout. However, a performance decrease was observed in complex situations where the accuracy and consistency of the segmentation and tracking tasks are low. In general, the recognition rate in controlled scenarios was higher, as expected, than in uncontrolled ones. Real-time operation was achieved in both situations. Furthermore, an in-depth study of the probability of the event detections showed that there is a high amount of events with intermediate values (e.g., $0.2 < score < 0.8$). Such values can be due to uncertainty of the low-level analysis or non-modeled situations.

As future work, we will investigate the inclusion of feedback-based analysis for studying events with intermediate probability as well as the application of the proposed approach to other domains.

References

References

[1] P. Turaga, R. Chellappa, V. Subrahmanian, O. Udrea, Machine recognition of human activities: A survey, IEEE Trans. on Circuits and Systems for Video Technology 18 (11) (2008) 1473–1488.

[2] D. Ayers, M. Shah, Monitoring human behavior from video taken in an office environment, Image and Vision Computing 19 (12) (2001) 833–846.

[3] F. Fusier, V. Valentin, F. Bremond, M. Thonnat, M. Borg, D. Thirde, J. Ferryman, Video understanding for complex activity recognition, Machine Vision and Applications 18 (3) (2007) 167–188.

[4] G. Lavee, E. Rivlin, M. Rudzsky, Understanding video events: A survey of methods for automatic interpretation of semantic occurences in video, IEEE Trans. on Systems, Man and Cybernetics - Part C: Applications and reviews 39 (5) (2009) 489–504.

[5] G. Lavee, M. Rudzsky, E. Rivlin, A. Borzin, Video event modeling and recognition in generalized stochastic petri nets, IEEE Trans. on Circuits and Systems for Video Technology 20 (1) (2010) 102–118.

[6] M. Albanese, R. Chellappa, V. Moscato, A. Picariello, V. S. Subrahmanian, P. Turaga, O. Udrea, A constrained probabilistic petri net framework for human activity detection in video, IEEE Trans. on Multimedia 10 (8) (2008) 1429–443.

[7] M. Ryoo, J. Aggarwal, Semantic representation and recognition of continued and recursive human activities, Int. Journal on Computer Vision 82 (1) (2009) 1–24.

[8] J. SanMiguel, M. Escudero-Vinolo, J. Martinez, J. Bescos, Real-time single-view video event recognition in controlled environments, in: Proc. of the Int. Workshop on Content-Based Multimedia Indexing, 2011, pp. 91–96.

[9] J. SanMiguel, J. Martinez, A. Garcia, An ontology for event detection and its application in surveillance video, in: Proc. of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance, 2009, pp. 220 –225.

[10] J. K. Aggarwal, M. S. Ryoo, Human activity analysis: A review, ACM Computing Surveys 43 (3) (2011) 1–43.

[11] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Computer Vision and Image Understanding 115 (2) (2011) 224–241.

[12] Q. Luo, X. Kong, G. Zeng, J. Fan, Human action detection via boosted local motion histograms, Machine Vision and Applications 21 (3) (2010) 377–389.

[13] W. Huang, Q. Wu, Human action recognition based on self organizing map, in: Proc. of IEEE Int. Conf. on Acoustics Speech and Signal Processing, Dallas (USA), 2010, pp. 2130 –2133.

[14] S. Hongeng, R. Nevatia, F. Bremond, Video-based event recognition: activity representation and probabilistic recognition methods, Computer Vision and Image Understanding 96 (2) (2004) 129–162.

[15] R. Martinez-Tomas, M. Rincon, M. Bachiller, J. Mira, On the correspondence between objects and events for the diagnosis of situations in visual surveillance tasks, Pattern Recognition Letters 29 (8) (2008) 1117–1135.

[16] N. Ghanem, D. DeMenthon, D. Doermann, L. Davis, Representation and recognition of events in surveillance video using petri nets, in: IEEE Int. Conf. on Computer Vision and Pattern Recognition Workshop, Washington DC (USA), 2004, pp. 104–112.

[17] S. Park, J. K. Aggarwal, A hierarchical bayesian network for event recognition of human actions and interactions, Multimedia Systems 10 (2004) 164–179.

[18] C. Achard, X. Qu, A. Mokhber, M. Milgram, A novel approach for recognition of human actions with semi-global features, Machine Vision and Applications 19 (1) (2008) 27–34.

[19] N. Cuntoor, B. Yegnanarayana, R. Chellappa, Interpretation of state sequences in hmm for activity representation, in: Proc. of IEEE Int. Conf. Acoustics, Speech and Signal Processing, Vol. 2, Philadelphia (USA), 2005, pp. 709–712.

[20] C. Town, Ontological inference for image and video analysis, Machine Vision and Applications 17 (2) (2006) 94–115.

[21] A. Bobick, A. D. Wilson, A state-based approach to the representation and recognition of gesture, IEEE Trans. on Pattern Analysis and Machine Intelligence 19 (12) (1997) 1325–1337.

[22] M. Balcells, D. DeMenthon, D. Doermann, An appearance-based approach for consistent labeling of humans and objects in video, Pattern Analysis and Applications 7 (1) (2005) 373–385.

[23] U. Akdemir, P. Turaga, R. Chellappa, An ontology based approach for activity recognition from video, in: Proc. of the ACM Int. Conf. on Multimedia, Vancouver (Canada), 2008, pp. 709–712.

[24] Y. Ivanov, A. Bobick, Recognition of visual activities and interactions by stochastic parsing, IEEE Trans. on Pattern Analysis and Machine Intelligence 22 (8) (2000) 852–872.

[25] G. Lavee, M. Rudzsky, E. Rivlin, Propagating uncertainty in petri nets for activity recognition, in: Proc. of Int. Symposium on Advances in Visual Computing, Las Vegas (USA), 2010, pp. 706–715.

[26] R. Romdhane, F. Bremond, M. Thonnat, A framework dealing with uncertainty for complex event recognition, in: Proc. of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance, Boston (USA), 2010, pp. 392–399.

[27] G. Lavee, A. Borzin, M. Rudzsky, E. Rivlin, Building petri nets from video event ontologies, in: Proc. of Int. Symposium on Advances in Visual Computing, Lake Tahoe (USA), 2007, pp. 442–451.

[28] R. Vezzani, R. Cucchiara, Annotation collection and online performance evaluation for video surveillance: The visor project, in: Proc. of the IEEE Int. Conf. on Advanced Video and Signal Based Surveillance, Santa Fe (USA), 2008, pp. 227–234.

[29] F. Lv, X. Song, V. Wu, B.and Kumar, R. Nevatia, Left luggage detection using bayesian inference, in: Proc. of IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance, New York (USA), 2006, pp. 83–90.

[30] J. Allen, Actions and events in interval temporal logic, Journal of Logic and computation 4 (5) (1994) 531–579.

[31] Y. Tian, R. Feris, H. Lui, A. Humpapur, M. Sun, Robust detection of abandoned and removed objects in complex surveillance videos, IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews 41 (5) (2010) 565–576.

[32] J. San Miguel, J. Martinez, Robust unattended and stolen object detection by fusing simple algorithms, in: Proc. of IEEE Int. Conf. on Advanced Video and Signal based Surveillance, Santa Fe (USA), ISSN=„ 2008, pp. 18–25.

[33] D. Doermann, D. Mihalcik, Tools and techniques for video performances evaluation, in: Proc. of IEEE Int. Conf. on Pattern Recognition, Cambrigde (UK), 2000, pp. 167–170.

[34] S. Y. Elhabian, K. M. El-Sayed, S. H. Ahmed, Moving object detection in spatial domain using background removal techniques - state-of-art, Recent Patents on Computer Science 1 (1) (2008) 32–54.