An Ontology for Event Detection and its Application in Surveillance Video

Juan Carlos SanMiguel, José M. Martínez, Álvaro García
Video Processing and Understanding Lab
Escuela Politécnica Superior, Universidad Autónoma de Madrid, SPAIN
E-mail: {JuanCarlos.SanMiguel, JoseM.Martinez, Alvaro.GarciaMartin}@uam.es

Abstract

In this paper, we propose an ontology for representing the prior knowledge related to video event analysis. It is composed of two types of knowledge related to the application domain and the analysis system. Domain knowledge involves all the high level semantic concepts in the context of each examined domain (objects, events, context...) whilst system knowledge involves the capabilities of the analysis system (algorithms, reactions to events...). The proposed ontology has been structured in two parts: the basic ontology (composed of the basic concepts and their specializations) and the domain-specific extensions. Additionally, a video analysis framework based on the proposed ontology is defined for the analysis of different application domains showing the potential use of the proposed ontology. In order to show the real applicability of the proposed ontology, it is specialized for the Underground video-surveillance domain showing some results that demonstrate the usability and effectiveness of the proposed ontology.

1. Introduction

Recently, the use of ontologies for representing the prior knowledge of a domain has been proposed in the video analysis research community [1]. In each domain, the determination of the number and type of the objects that can appear in the scene allows to link the visual analysis process with the specific domain ontologies (that is, descriptions of possibly detectable objects, relationships between objects...). This approach is currently obtaining promising results [2][3][4]. Moreover, the ontology is useful to evaluate scene understanding systems, to understand exactly what types of events a particular system can recognize, and to share and reuse models dedicated to the recognition of specific events.

In this paper we propose an ontology that integrates two kinds of knowledge information: the scene and the system. Scene knowledge is described in terms of objects, relations between them (events), spatial context... showing how scene objects and simple or complex events can be described. System knowledge allows determining the best configuration of the processing schemes for detecting the objects/events of the scene by describing its capabilities (e.g., analysis algorithms), its reactions to specific events and different analysis schemes. The contribution of this paper is the integration of different types of knowledge in an ontology with the purpose of detecting the objects and events in a video scene. In addition, we show how the complex events can be described by simple events and we define a simple procedure to build a visual analysis framework using the proposed ontology.

The paper is organized as follows: section 2 overviews the state-of-the-art in knowledge representation using ontologies, sections 3 and 4 describe the core ontology and the specializations for the video surveillance domain, section 5 presents a video analysis framework that makes use of the proposed ontology, section 6 describes a case of study (for the Underground Video-surveillance domain) and finally section 7 raises some conclusions and proposes lines of future work.

2. Knowledge representation using ontologies

The use of ontologies for representing the contextual information has caused the development of knowledge-based systems [3]. These systems have to deal with two basic problems: how to build and represent the entities of the domain modeled in the ontology, and how to use the ontology for improving the video analysis results.

For solving the first problem, most of the proposals define the components and the ontology structure in a manual way (e.g. [3][4][5]), obtaining the low level attributes by extracting visual descriptors from available training sequences. Commonly, a generic ontology is defined for the analysis problems and specific domain ontologies are defined for each application domain considered. The different proposals represent their ontologies with formal languages usually employed in Artificial Intelligence and Semantic Web for knowledge representation. For example, in [5][6] RDFS (Resource Description Framework Schema) is used, whilst in [3] the choice is DL (Description Logic). VERL, an ontology

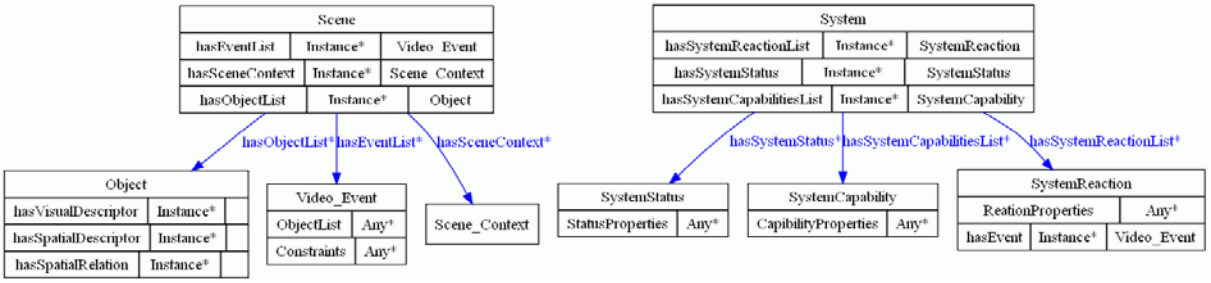

Figure 1: OntoViz visualization of the Scene and System basic classes of the ontology

representation language for describing events in video sequences based on OWL [2][7], has been proposed as an initiative for dynamizing the use of ontologies in all the areas of video processing.

The solutions taken to solve the second problem are more heterogeneous. For example, the analysis process is specified in [6] by using a set of logic rules for inference expressed in F-Logic format. In [8], the event detection is performed using a set of rules in SWRL language. In [5] a genetic algorithm is proposed to find the optimum matching between the ontology entities and a set of regions initially extracted automatically from the images. Additionally, in [4] an ontology is used for defining a Dynamic Bayesian Network by applying a training phase with the aim of identifying the entities defined.

3. Basic ontology

3.1. Basic entities

In a generic video analysis system, we can suppose that the prior knowledge is composed of two parts: the scene (or the outside) and the system (or the inside). Thus, the basic ontology is defined by two main entities: *Scene* and *System*. The *Scene* entity describes the physical space where a real event occurs and which can be observed by one or several cameras. It includes a list of the scene objects, a list of their interactions (events) and any other a priori information acquired before the processing of the scene (context of the scene). The *System* entity concept represents the system that uses the ontology including its analysis capabilities, the possible responses to the detected events, its status and the processing schemes for achieving different objectives. Fig.1 depicts the *Scene* and *System* basic entities and their relation with the ontology specializations.

3.2. Ontology specializations

3.2.1 Object specializations

The *Object* concept represents any real world object observed by the camera. It is the basic structural unit in the scene and most of the ontology concepts (*SceneContext*, *Event*,...) rely on the *Object* entity. The class of an object corresponds to its nature and can be determined by its properties. The objects in the scene can be classified depending on their ability to initiate their own motion. Thus, *Mobile Objects* are objects that can initiate its motion and *Contextual Objects* are objects that cannot initiate its motion. Depending if the object is movable, we have two subclasses of *Contextual Objects*: *Fixed Objects* (if they cannot be displaced) and *Portable Objects* (if they can be displaced). Furthermore, scene objects can be classified depending on the model dimensionality: 2D or 3D.

The attributes of an object are all the properties characterizing the *Object* entity. These properties are divided in two types: inter-object properties (like "isPartOf" and "hasSpatialRelation") and intra-object properties (basic visual attributes: local/global-appearance and MPEG-7 Visual Descriptors). Additionally, each *Object* entity has a property that indicates the detection procedure for the entity ("hasDetectionProcedure").

Fig 2 depicts a *Mobile Object* entity example for the entity that defines a person.

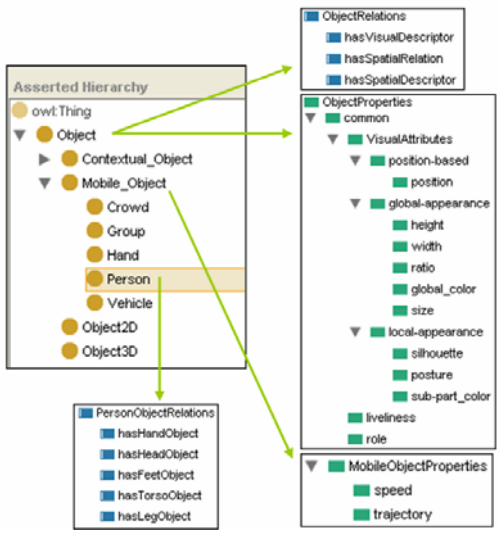

Figure 2: Ontology model for the Person entity (inter/intra object attributes are indicated in blue/green colour respectively)

3.2.2 *Event specializations*

The *Event* entity is a generic term to describe any event, action or activity that happens in the scene and that is interesting for a video analysis system (e.g. video surveillance, video indexing). The semantic level of an event is very variable: from low-level events (e.g., motion, illumination changes), mid-level events (like appearing, splitting and other object-related events) or high-level events (e.g., gesture identification, activity monitoring). In the proposed ontology we define the high-level video events by describing their objects of interest and the relations between them.

Three different aspects for classifying the video events are proposed: number of objects of interest (*Single* or *Multiple*), temporal relation (*Simple* for short-term events or *Complex* for long-term events divided into sub-events) and transitivity (*Intransitive* or *Transitive*).

In the rest of the paper, the transitivity-based event classification is not detailed explicitly as it only adds the features *action_subject* and *action_purpose* to all the modeled events. Thus, combining the two first aspects we have the following event types: *Simple_SingleObject* events (SSE), *Simple_MultipleObject* events (SME), *Complex_SingleObject* events (CSE) and *Complex_MultipleObject* events (CME).

The attributes of an event are all the properties characterizing the *Event* entity. These properties are divided into a list of objects that perform the action (*ObjectList*), a list of sub-events that compose the event to detect (*Sub-events*), the relations or constraints between the objects, the sub-events that compose the event to be modeled (*Constraints*), and the detection procedures of each sub-event (*DetectionProcedureList*)

3.2.3 *Context specializations*

The *SceneContext* entity defines all the information that may influence the way a scene is perceived. This information can be used during the analysis of the scene to help the process to complete the task efficiently (e.g., event detection).

In this context, we distinguish three types of a priori useful information: *SpatialContext* (a spatial description of the scene), *ObjectContext* (domain-specific relations between objects) and *EventContext* (domain-specific relations between events: the more likely events, combinations among the events, spatial location…).

3.2.4 *System Capabilities specializations*

The *System Capabilities* entity defines all the available resources in the system that analyzes the video content.

In this section, we distinguish two different types of system capabilities: the algorithms and the detection procedures. The *Algorithm* entity describes all the available processing algorithms in the system and the

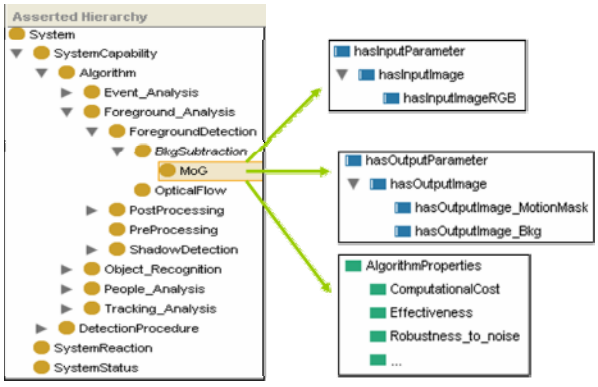

Figure 3: Ontology model for the *Algorithm* entity

DetectionProcedure entity describes the relations between the concepts or entities susceptible of being detected (like the objects, events…) and the available algorithms. This relation includes the different processing schemes available to complete a specific detection task.

The attributes of an algorithm are all the properties characterizing the *Algorithm* entity. These properties are divided into a list of inputs (*InputParameterList*), a list of outputs (*OutputParameterList*) and a list of properties for classifying the performance of the algorithm (like computational cost, effectiveness…). This list of properties is subjectively obtained by evaluating each algorithm with a specific dataset for the corresponding task. An entity has been defined (*Parameter* entity) for describing the different inputs and outputs of the algorithms available in the system. The specializations of this entity are used as the elements in the input/output lists of each *Algorithm* entity. Fig. 3 depicts an example of the *Algorithm* entity.

The attributes of a detection procedure are a list of algorithms used in the detection (*AlgorithmList*), the usage order of the algorithms (*OrderList*) and the entity analyzed in the detection procedure (*Object* or *Event*). The list of inputs and outputs of the detection procedure can be inferred from the properties of the algorithms involved.

3.2.5 *System reactions specializations*

This part of the ontology defines the different system reactions to specific detected events or objects (like starting another application, recording the event to disk...).

In the proposed ontology, we distinguish four different types of system reactions: reactions to detected events (like display an alarm when an event occurs or record some event to a video file in disk), reactions to system failures (for example, decide which action will be done if some component of the system fails), reactions to user events (user interaction with the interface of the application) and other type of reactions.

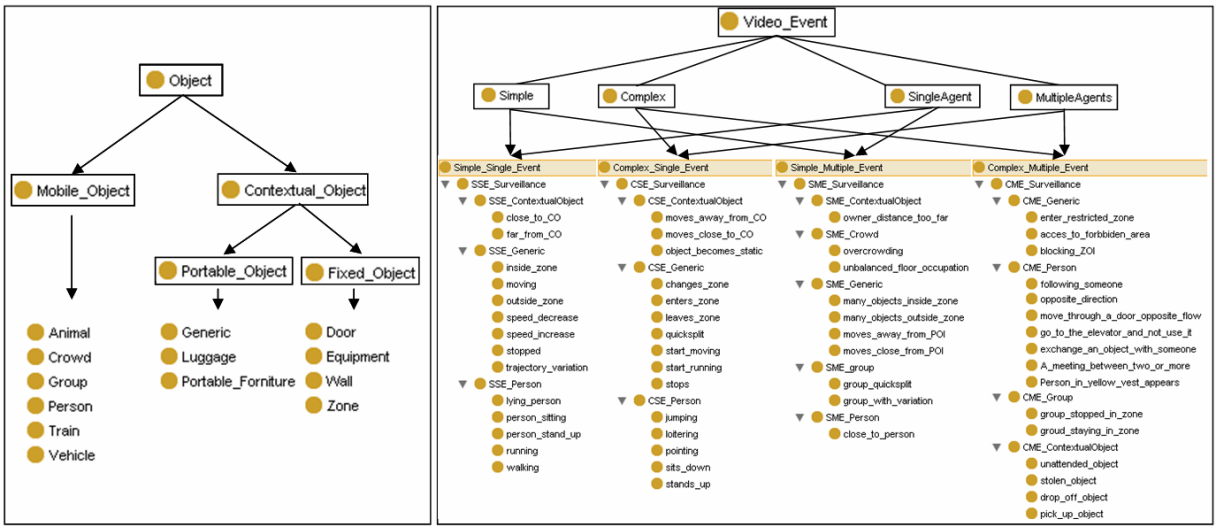

Figure 4: Object and Event class specializations and detailed hierarchy for the Underground Video-surveillance domain

4. Domain-specific Specializations

4.1. Underground Video-surveillance domain

In this section, we describe a domain-specific specialization of the proposed ontology framework, namely for the Underground Video-surveillance domain.

4.1.1 Object specializations

For the *Object* concept, only a limited number of object types are observed in Underground video-surveillance sequences: person, group of persons, metro train, portable objects,... Low-level features (see Fig. 2) are used to discriminate between the different domain objects. A description of the objects modeled is depicted in Fig. 4.

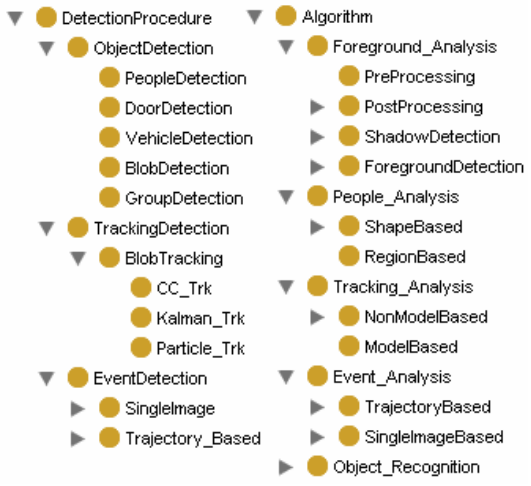

Figure 5: Description of the *Algorithm* and *DetectionProcedure* entities for the Underground video-surveillance domain

4.1.2 Event specializations

For the *Event* concept, we have described the events of interest related to the domain being modeled. These events have been classified depending on the previously described categories (see Fig. 4). This classification allows the application of different event detection models for each category depending on the available algorithms.

4.1.3 Context specializations

For the *SceneContext* concept, we have only described the *SpatialContext* related to the scene being monitored. This context includes the annotation of the fixed objects of the scene (zones of interest, train tracks, doors, walls, seat...) and their spatial location.

4.1.4 System capabilities specializations

For the *Algorithm* entity, we have described the common algorithms used in the analysis of Underground video-surveillance sequences. It is divided into the different stages that can be used and they are: foreground analysis (divided into pre/post-processing methods, shadow detection methods and foreground detection methods), tracking analysis (divided into connected components, Kalman filter and particle filter for blobs or specific objects), people detection (divided into shape-based or region-based methods) and event analysis routines (divided into trajectory-based and single image-based). For the *DetectionProcedure* entity, we have described detection schemes for the detection of blobs or people, the tracking of blobs or people, and the detection of some events or sub-events (like “people inside a zone”, “get/put objects” or “abandoned/stolen object”). A description of the specializations of the *Algorithm* and *DetectionProcedure* entities is depicted in Fig. 5.

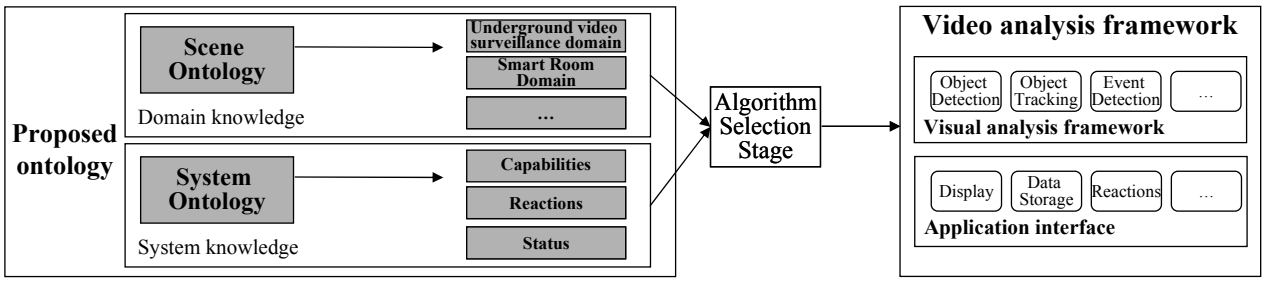

Figure 6: Overview of the proposed framework

5. Framework for Video Surveillance Analysis Based on the proposed ontology

According to the available knowledge described in the ontology, a framework has been designed and implemented supporting the described specific domain. It is depicted in Fig. 6 and it can be extended to other domains by developing the respective domain ontology. All the shared knowledge (domain-specific specializations for object and event entities, available spatial context, processing schemes and algorithms,...) is combined in a selection stage to determine the best visual analysis framework (or processing scheme), user interfaces and system reactions to the analysis of the modeled domain.

As an initial implementation of this selection stage for building the visual analysis framework, we have decided to process the events defined in the domain ontology in three phases to determine the best visual analysis framework. Firstly, the sub-events that compose each defined event of the ontology are analyzed and their corresponding detection method is selected (using the "hasDetectionProcedure" property). Secondly, each selected event detection procedure is inspected to determine its inputs and the algorithms used in the detection procedure. For example, the "*PeopleDetection*" procedure uses a *foregroundDetection*, a *ShadowDetection* and a *NoiseRemoval* stages to detect people. Thirdly, all the algorithms used in the different detection procedures selected are listed and the repeated ones are eliminated from this list (avoiding the inclusion of repeated algorithms). Finally, this list of algorithms is used to build the visual analysis framework that performs the event detection task. Additionally, a supervisor module is introduced in the analysis scheme to interconnect the different algorithms or processing stages (providing the needed input parameters and storing the output parameters) and to manage detection procedures/schemes used (forward or feedback configuration schemes) and the detection models for each relevant concept (e.g., background subtraction for detecting foreground objects, Bayesian inference or state machines for detecting simple or complex events).

6. Case of Study

In this section, we propose as case of study the detection of events in the Underground video-surveillance domain using the proposed ontology.

The system has been implemented in C++, using the OpenCV image processing library (http://sourceforge.net/projects/opencv/) and integrated in the DiVA video processing system [9]. Tests were executed on a Pentium IV with a CPU frequency of 2.8 GHz and 1GB RAM. The Protégé software (http://protege.stanford.edu) has been used to create the proposed ontology using OWL as the ontology language.

Experiments were carried out on several sequences from the the i-LIDS dataset for AVSS2007 (available at http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007.html) and the PETS2006 dataset (available at http://www.pets2006.net/). Visual results and details about the selected sequences can be found at http://www-vpu.ii.uam.es/publications/OntologyForEventDetection.

6.1. Creation of the visual analysis framework

In this section, the creation of the visual analysis framework using the selection stage described in section 5 is presented. As a first step, a subset of events has been selected to test the proposed ontology. They are: *GetObject*, *PutObject*, *AbandonedObject* and *StolenObject* events.

In the selection stage, the detection procedures of each event are examined to determine the necessary processing stages. For example, *GetObject* needs a description of the blobs in the scene, their people likelihood and basic tracking information of them. Then, these procedures are stored in a list and they are examined to determine the algorithms to be used. For example, the *BlobDetection* procedure needs a foreground detection stage followed by a connected component analysis. *PeopleDetection* implies a *BlobDetection* procedure followed by a people analysis stage (using the people detection algorithms). All the detection procedures are recursively examined and the repeated algorithms are deleted. Fig. 7 depicts the visual analysis framework finally constructed.

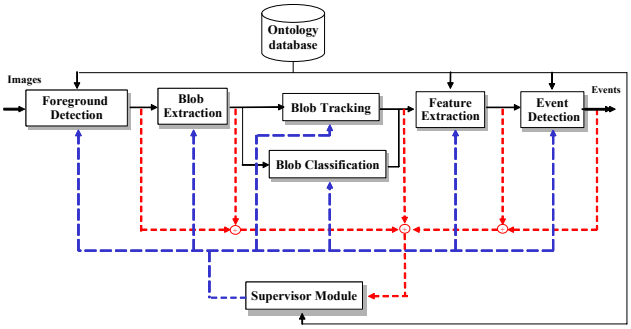

Figure 7: Visual analysis framework derived from the events defined in the proposed domain ontology

6.2. Performance evaluation

In this section we evaluate the performance of the developed visual analysis framework that makes use of the ontology.

Table 1 shows "Precision" and "Recall" for detecting the proposed subset of events. It can be observed that the performance measures are below state-of-art traditional approaches. This is mainly because the definition of events in the ontology is very strict and only slight differences between the definition and the occurrences are allowed. Using diffuse restrictions instead of strict logic ones defined by the ontology, will allow to increase the recall. On the other side, we are using quick algorithms in order to achieve a high processing rate (26,5 fps for CIF sequences), and therefore another direction of enhancement will be balancing the trade-off between processing time and efficiency of the algorithms in order to reach a compromise between both requirements.

Table 1: *Precision and recall for the real-time visual analysis framework developed.*

Event	Manual Annotations	Precision	Recall
GetObject	20	79%	67%
PutObject	33	77%	60%
Abandoned Object	12	74%	59%
Stolen Object	8	75%	62%

7. Conclusions and Future Work

In this paper, we have presented a formalization of knowledge relevant to video analysis systems using an ontology. The relevant information is described in terms of scene-related entities (*Object, Event*, and *Context*), system-related entities (*Capabilities, Reactions*...). Then, a domain extension is proposed for the Underground Video-surveillance domain. In addition, the knowledge described in the ontology has been mapped to a visual analysis framework dependant on the events defined to be detected. It demonstrates the potential use of this ontology to define complex events based on a list of simple events (using the detection procedures available in the system). Furthermore, the design of the ontology allows to easily define new processing schemes using the algorithms described in the ontology focusing the effort on the scheme design phase. Future work includes an extension of the ontology with more complex representations of the *Object* and *Event* entities and an extension of the selection stage to build the visual analysis framework supporting more complex rules. Additionally, the development of new domain ontologies (e.g., smart rooms domain), the inclusion of new processing schemes in the System part of the ontology (e.g., complex feedback strategies) and the inclusion of a reasoning stage to manage the failure of the system components will be explored.

8. Acknowledgments

This work is partially supported by the Spanish Administration agency CDTI (CENIT-VISION 2007-1007), by the Spanish Government (TEC2007- 65400 SemanticVideo), by the Comunidad de Madrid (S-050/TIC-0223 - ProMultiDis), by the Consejería de Educación of the Comunidad de Madrid and by The European Social Fund.

9. References

[1] Bremond, F.; Maillot, N.; Thonnat, M.; Vu, V.; "*Ontologies for video events*", Technical report nº 5189, INRIA, 2004.

[2] Francois, A.R.J.; Nevatia, R.; Hobbs, J.; Bolles, R.C.; Smith, J.R.; "*VERL: an ontology framework for representing and annotating video events*", IEEE Multimedia, 12(4):76-86, 2005.

[3] Maillot, N.; Thonnat, M.; Boucher, A.; "*Towards ontology-based cognitive vision*", Machine Vision and Applications, 16 (1): 33–40, 2004.

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

[5] Voisine, N.; Dasiopoulou, S.; Precioso, F.; Mezaris, V.; Kompatsiaris, I.; Strintzis, M.G.; "*A genetic algorithm-based approach to knowledge-assisted video analysis*," Proc. IEEE Conf. on Image Processing, 3:441-444. 2005

[6] Dasiopoulou, S.; Mezaris, V.; Kompatsiaris, I.; Papastathis, V.-K.; Strintzis, M.G.; "*Knowledge-assisted semantic video object detection*," IEEE Trans. on Circuits and Systems for Video Technology, 15(10):1210-1224, 2005.

[7] Bolles B.; Nevatia,R.; "*A hierarchical video event ontology in owl*," Proc. ARDA Challenge Workshop, 2004.

[8] Snidaro, L., Belluz, M., Foresti, G.L.;"*Representing and recognizing complex events in surveillance applications*", Proc. *IEEE Conf on Advanced Video and Signal Based Surveillance,* pp.493-498, 2007.

[9] San Miguel, J.C.; Bescos, J.; Martinez, J.M.; Garcia, A.; "*DiVA: A Distributed Video Analysis Framework Applied to Video-Surveillance Systems*", Proc. Int. Workshop on Image Analysis for Multimedia Interactive Services, pp.207-210, 2008.