

This document is published in:

Corchado, E. et al. (eds.) (2011). *Hybrid Artificial Intelligent Systems: 6th International Conference, HAIS 2011, Wroclaw, Poland, May 23-25, 2011, Proceedings, Part II*. (Lecture Notes in Computer Science, 6679) Springer, pp. 144- 154. DOI: [http://dx.doi.org/10.1007/978-3-642-21222-2\\_18](http://dx.doi.org/10.1007/978-3-642-21222-2_18)

© 2011 Springer-Verlag Berlin Heidelberg

# A General Purpose Context Reasoning Environment to Deal with Tracking Problems: An Ontology-Based Prototype

Miguel A. Serrano, Miguel A. Patricio, Jesús García, and José M. Molina

GIAA, Universidad Carlos III de Madrid, Avenida de la Universidad Carlos III, 22,  
Colmenarejo, Madrid, Spain

{miguel.serrano,miguelangel.patricio,jesus.garcia,  
josemanuel.molina}@uc3m.es

**Abstract.** The high complexity of semantics extraction with automatic video analysis has forced the researchers to the generalization of mixed approaches based on perceptual and context data. These mixed approaches do not usually take in account the advantages and benefits of the data fusion discipline. This paper presents a context reasoning environment to deal with general and specific tracking problems. The cornerstone of the environment is a symbolic architecture based on the Joint Directors of Laboratories fusion model. This architecture may build a symbolic data representation from any source, check the data consistency, create new knowledge and refine it through inference obtaining a higher understanding level of the scene and providing feedback to autocorrect the tracking errors. An ontology-based prototype has been developed to carry out experimental tests. The prototype has been proved against tracking analysis occlusion problems.

**Keywords:** Context Reasoning, Occlusion Problem, Knowledge Approach, Video Tracking.

## 1 Introduction

The widespread use of video applications in new areas and to new publics has led to an increase amount of raw and unclassified resources. This trend has generated the need to automatically get semantic information by analyzing the resources and cataloging the data to achieve a scene interpretation.

Low informative quality of the data presented by video applications has resulted in an incapability to manage the extracted information on a semantic level. For years researchers have been working in two different ways to save the semantic gap; machine learning methods focused on low level visual descriptors such as texture, shape, etc. and knowledge approaches, which try to extract semantic descriptions using a higher abstraction perspective. None of these approaches can only by itself cover the entire problem.

Increasingly low level and knowledge approaches are being used in a synergistic way taking in account their abstraction level. Machine learning methods are used to imitate the way users assess visual similarity. Knowledge-based approaches are used

to discover new hidden domain knowledge through inference operations. However, these mixed approaches are not normally prepared to accept different kinds of measures from different kinds of devices, to fuse all the data generated and to make inference reasoning over these. In addition, normally with these solutions is not possible to deal specific problems at different abstraction levels.

In order to solve these limitations this paper presents an integral environment which combines machine learning methods with a general purpose knowledge approach based on symbolic representation formalism. The goal of this environment is to carry out a comprehensive tracking analysis extracting all the semantics at different abstraction levels from the resources.

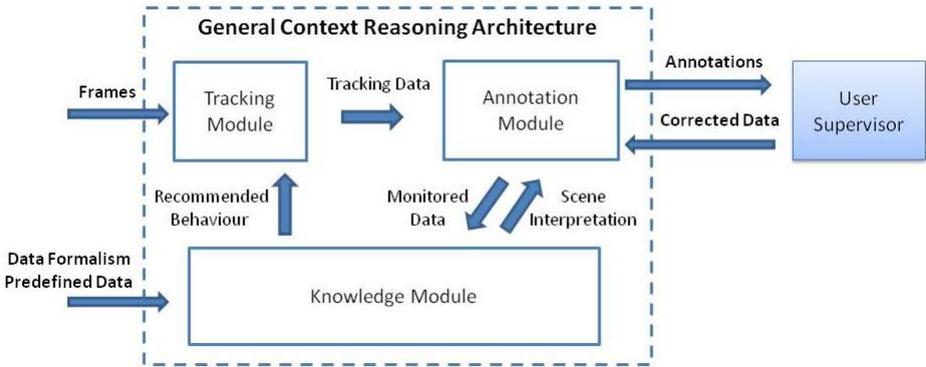
The general context reasoning environment is divided in three main modules a tracking system, an annotation system and a knowledge system. The tracking module collects part of the data performing a low level analysis over the raw resources. The annotation module carry out tasks such as annotating and showing the attained and inferred data, controlling the main flow of the execution and acting as interface between the user and the rest of the system. The knowledge module is built under the guidance of an architecture [3] based on a symbolic version of the Joint Directors of Laboratories (JDL) model [4] to deal with tracking problems. The JDL-based architecture is the cornerstone of the overall architecture. Their main capabilities are symbolic representation of the real world with ontologies, information flow refined through deductive and abductive reasoning from low level data to high level domain knowledge, consistency in the data inherent to the chosen representation formalism and feedback information that makes suggestions to improve the behavior of the tracking system.

To test the potential of this environment a prototype has been developed. As a demonstration of the current performance to solve problems at several levels, the system has been tested to the automatic detection and treatment of occlusion situations between scene objects.

The paper is organized as follows. In Section 2 the overall architecture is presented. Section 3 shows the detailed implementation of knowledge model prototype. Section 4 shows an example of how a typical video tracking problem is solved applying the solution presented. Section 5 explains the conclusions obtained and the future work.

## 2 Overall Architecture

As it is illustrated in Fig. 1, the overall architecture is composed by a tracking module, an annotation module and a knowledge module. A user supervisor is normally necessary to monitor the tracking performance. The basic inputs for this architecture are a data formalism, a variable quantity of predefined data and the video frames. The two first, are inputs of the knowledge module. The data formalism represents the skeleton of concepts, placed according to their abstraction level, where the data is stored. Optionally this formalism could be accompanied with predefined data, such as, assertions of a knowledge base. Normally these data come from previous analysis and context information. The video frames are the tracking module inputs; each frame starts the sequence analysis whose information stream covers all the environment elements.

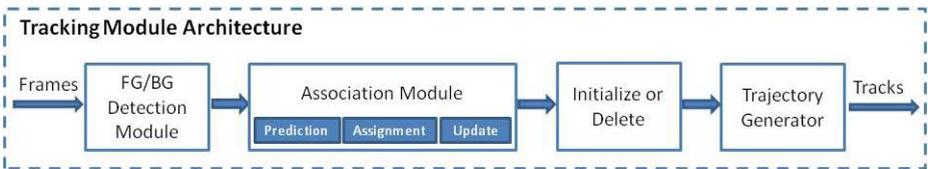


**Fig. 1.** Overall architecture of the general context reasoning environment

After a frame entry, a real time tracking analysis is performed. When the analysis have finished the tracking module returns information related with tracking entities, such as, position, size, etc. The annotation module supports two inputs the tracking data and the previous frame scene interpretation. The scene interpretations are the semantic conclusions of the knowledge module about the situation of the video scene at different abstraction levels. All these inputs are presented as annotations to the user supervisor. Once the information is checked by the user, the annotation module sends this data monitored to the knowledge module. The knowledge module triggers its reasoning abilities obtaining a new interpretation of the video scene and recommendations of how should the tracking module behave during the analysis.

## 2.1 Tracking Module

The tracking module architecture, presented in Fig 2, is based on a video chain with different submodules that run in sequence, which correspond to the successive phases of the tracking process.



**Fig. 2.** Tracking Module Architecture

This architecture is composed by four processes: Foreground/Background Detection module, shows when a pixel has moved and group them in blobs. Association module, predicts the blobs positions, assign sets of blobs to tracks and finally update the tracks positions. Initialize or Delete module, create and delete tracks when have not assigned to any blob. Trajectory Generator module, detects anomalous behaviors studying tracks trajectories. The algorithms belonging to each submodule are interchangeable in each run. The input data for the pipeline is a frame and the output data is the track features.

## 2.2 Annotation Module

The annotation module is based on the ViPER-GT tool [10]. The initial goal of this tool was to provide instruments to easily create and share ground truths in a flexible format. It was designed to allow frame-by-frame markup of video metadata stored in a XML format. The GUI developed is very useful for visualization and could be used to record the requisite information in a single scan of the video content.

The modified version of this tool [7] allows user to supervise data generation through predefined dynamic templates. Predefined templates present the information related with the knowledge module abstraction levels tracking, objects, activities, etc. The annotation tool also offers some possibilities like creation, deletion and update on the semantics of the scene, for example, the user could update in a template the size of a track. All the new data have to be finally sent to the knowledge module due to user updates may modify the interpretation of the global situation. For example, the user could update in a template the size of a context object which modifies the semantic relationships between some other objects.

## 2.3 Knowledge Module

The knowledge module is an ontology based model compliant with JDL to represent knowledge in cognitive visual data fusion systems [3]. This model is stepped in several levels from low-level track data to high level situations whose structure is determined by a set of ontologies. Each ontology level provides a skeleton that includes general concepts and relations to describe the computer vision entities and their relations.

- TREN ontology (L1). The input data come from the tracking module and represent tracked entities information (color, position, velocity, etc.) and frames.
- SCOB ontology (L1-L2). Represents real-world entities, properties and relations of the scene: moving and static objects, topological relations, etc.
- ACTV ontology (L2). Models the behavior descriptions: grouping, approaching, picking an object, etc.
- IMPC ontology (L3). Define the association of a cost value to activity descriptions.

The ontology model has been designed to promote extensibility and modularity. This general structure might be refined to apply this model in a specific domain. It is interesting to note that the amount of data in lower level ontologies is larger than higher levels. Concepts that belong to less abstract knowledge form the basis of the high abstract knowledge.

These ontologies can contain both perceptual and context data. Perceptual data is automatically extracted by the tracking algorithm. Context data is external knowledge used to complete the comprehension of the scene. Context data includes information about scene environment, the parameters of the recording, information previously computed and user-requested information. For example, an static object and its features size, position, topological state, etc. are regarded as context data. The

producing sources of context are the a priori knowledge and the users. A priori knowledge is generally common sense data introduced before the initialization of the system. This information is usually closely linked to the specific purpose of the application. Users must also provide the required amount of the context data interacting with the annotation module. This required data should be introduced during the execution when a bad comprehension of the scene is detected in the behavior of the tracking module.

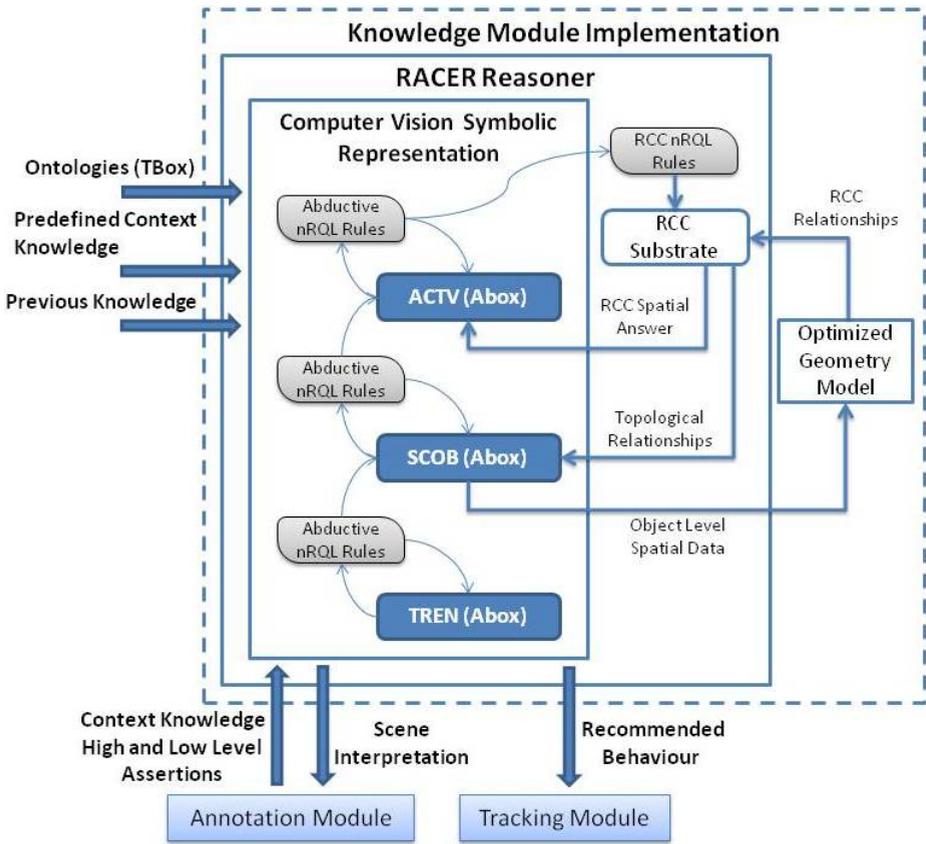
New knowledge assertions in each ontology from the proper and the previous level are also possible thanks to the abductive reasoning. Beyond the standard reasoning based on the subsumption ontology mechanism, this model can perform rule based inferences using a description logic inference engine. Deductive and abductive rules are the two main types of reasoning procedures. Deductive rules maintain the consistency in the ontology. The consequent include logical conditions and not include in the consequent any individual not mentioned in the antecedent. Abductive rules are used to achieve the scene interpretation and creation of feedback for the tracking module. These rules allow the inclusion of additional individuals in the consequent, which are created as new instances of the ontology. Abductive reasoning use additional mechanism like data retrieval queries, previously stored queries or queries depending on arithmetical calculus.

### **3 Ontology Based Prototype: Knowledge System**

The ontology based prototype implements the overall architecture seen in section 2, however, in terms of research quality the key piece of this architecture is the knowledge module. This module must carry out the fundamental tasks such as asserting symbolic context data, creating a symbolic representation of these data, checking the data consistency, making deductive and abductive inference and providing feedback at different levels.

As is illustrated in Fig. 3, the developing of this module is almost entirely based on the RACER reasoner [9]. RACER has been selected because of the necessity of the system to develop different kinds of reasoning throw rules capable to infer new knowledge. In addition, RACER is one of the first reasoners to treat the spatial knowledge with an implementation of the Region Connection Calculus (RCC) theory.

The knowledge module is represented by the computer vision symbolic representation (CVSR). The actual CVST implementation includes the three first lowest levels of the architecture, the tracking entities level (TREN), the scene object level (SCOB) and the activities level (ACTV). These levels are defined from a set of terminological box (TBox) that contains the hierarchical description of the concepts defining the classes and relations of the ontology of each level. CVSR assertional boxes (ABoxes) indicate the assertions about the individuals in the domain. The consistency verify whether all concepts in the TBox admit at least one individual in the corresponding ABox. These individual assertions come from predefined context knowledge given by the user or previous executions. Currently the system loads all



**Fig. 3.** The knowledge module implementation of the ontology based prototype

these data, prior to the start of the execution, from a file previously saved with information of an ABox.

During the execution, more context knowledge could be asserted by the user through the annotation system interface or by abductive new Racer Query Language (nRQL) rules. nRQL can be used to query RACER ABoxes, TBoxes and substrates and RDF and OWL documents. Abductive rules make flow new and more elaborated data at the same level or from low to upper abstraction levels. In the example below it can be seen a rule that generate new upper knowledge transforming a track in a dynamic scene object of the person class. This rule makes use of some reasoning facilities previously explained. Rule antecedent contains perceptual data such as track identifiers, previously stored queries like current-snapshot function, the a priori context knowledge showed in the width and height track properties condition, etc. Rule consequent generates a new individual labeled as person, declares a specific identifier and creates a new role which associates the individual to a track entity.

```

(firerule
  (and
    (?track (equal #!tren:id trackID)
    (?track ?tsnapshot ?dimension current-snapshot)
    (?tsnapshot #!tren:unknown_frame #!tren:isValidInEnd)
    (?dimension (>= #!tren:width 9))
    (?dimension (>= #!tren:height 9)))
  (
    (instance (new-ind |ind|) #!scob:Person)
    (instance (individual |ind|)(equal #!scob:id trackID))
    (related (individual |ind|)?track #!scob:hasAssociatedTrack)))

```

Abductive reasoning with spatial objects is very expensive in terms of development of rules and time computing and grows with the increasing number of entities and the complexity of the scene. An ontology-centric architecture based on an optimized geometry model [8] has been designed to solve the topological relationships between spatial objects automatically. The capability for automatic assertion is given by an object model based on geometries. The object model seeks to prioritize the optimization using a dynamic data structure of spatial data. The main goal is the automatic storage of the spatial relationships on a RCC without a noticeable loss of efficiency.

RCC is a theory used for qualitative spatial representation and reasoning. This theory provides a formalism which allows inferring implicit knowledge from explicit knowledge and a set of qualitative spatial relationships to describe the relative location of spatial objects to one another. The RCC theory is implemented by RACER as a substrate. In general substrate representation layers are used to associate a RACER ABox with an additional representation layer. The RCC substrate is a special kind of data substrate used to represent domain objects that also have spatial characteristics. This substrate also offers querying facilities, nRQL support both spatial and combined spatial and non-spatial queries. A significant amount of knowledge from the scene object and activities level is deducted from rules that manage the spatial object relationships discovered by the optimized geometries model.

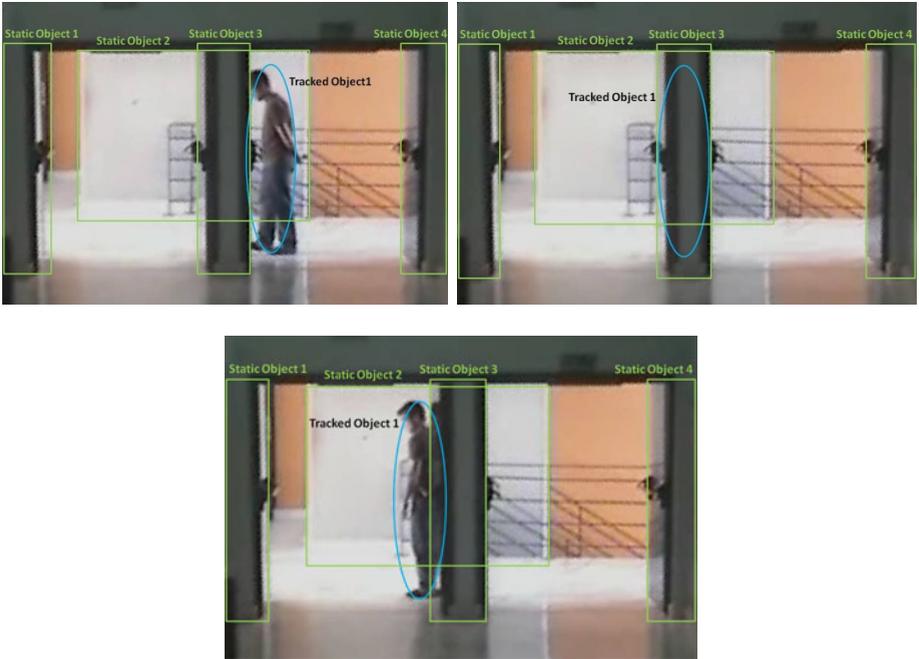
The scene interpretation output is presented to the user at the predefined dynamic templates of the annotation system. This allows the user to edit the erroneous semantic conclusions of the knowledge module at different abstraction levels to improve further recommendations. The tracking system recommendations have not been fully adapted to the prototype. The architecture presented in [1] and [2] uses a specific ontology called RECO which abstractly represents actions to improve the accuracy of the tracking system, and consequently, the quality of the measures of the fusion system. The current development directly modifies the features of the tracks at the lowest level to change the tracking behavior.

## 4 Case Study: Occlusion Problem

In this case study the main aim is to recognize and treat occlusion situations. The system has been tested against videos which show a scenario with several columns.

These columns have been defined by the user supervisor as context information and more specifically as a column concept which may be also considered as a static objects because one is a less general than the other.

**Table 1.** Illustrated sequence of how the overall system treats the occlusion. 1A shows the beginning of an occlusion. 1B shows the track completely occluded by the column. 1C image shows the occlusion ending.



The information given from the user to the system was the position, the size and the definition of the columns. The tracking system automatically detects the track position at every moment. This track is asserted in the TREN ontology with its features. The knowledge module detect that the track is a person and creates an individual in SCOB to represent the track at the object level from the rule view in section 3. Thus the system can make inference on the same level with columns and tracks.

An occlusion is detected when a tracked object in an overlap relation with a static object, in this case a column, is deleted by the tracking system. In 1A the prototype does not confuse as an occlusion the situation between the “static object 2” and the “tracked object 1” because the tracked object is not deleted by the tracking system. To recognize and assert occlusion situations is necessary the use of inference rules with tracking entities and scene objects in the antecedent and activities in the consequent.

```

(firerule
  (and
    (?track #!tren:Track)
    (?object #!scob:StaticObject)
    (?track (equal #!tren:id trackID))
    (?track ?person #!scob:isAssociatedToObject)
    (?*object ?*track :po))
  (
    (instance (new-ind |ind|) #!actv:Occlusion)
    (related (individual |ind|) ?object #!actv:hasOccluder)
    (related (individual |ind|) ?person #!actv:hasOccluded))

```

**Table 2.** Knowledge management sequence corresponding to the the image sequence in table 1

	<b>Image 1</b>	
<b>Previous Knowledge</b>	TREN	<b>Instances:</b> Track_1 [position and size]
	SCOB	<b>Instances:</b> Static_object1, Static_object2, Static_object3, Static_object4 [positions and sizes]
<b>Rule fired</b>	Track_1 matches the person rule, new person instance is created.	
<b>Rule Asserted Knowledge</b>	TREN	<b>Relationships:</b> Track_1 isAssociatedTo Object PersonObj1
	SCOB	<b>Instances:</b> PersonObj_1 <b>Relationships:</b> PersonObj_1 hasAssociatedTrack Track_1
<b>Tracking Event</b>	Track_1 deleted.	
<b>Knowledge Event</b>	Track_1 and PersonObj_1 declared as inactive in knowledge base. Overlapping rule fired.	
	<b>Image 2</b>	
<b>Previous Knowledge</b>	SCOB	<b>Instances:</b> Static_object1, Static_object2, Static_object3, Static_object4 [positions and sizes]
<b>Rule fired</b>	PersonObj_1 matches the overlapping rule.	
<b>Recommended Asserted Knowledge</b>	TREN	<b>Instances:</b> Track_1 [new position, old size] <b>Relationships:</b> Track_1 isAssociatedTo Object PersonObj1
	SCOB	<b>Instances:</b> PersonObj_1 <b>Relationships:</b> PersonObj_1 hasAssociatedTrack Track_1
<b>Rule Asserted Knowledge</b>	ACTV	<b>Instances:</b> Occlusion_1_3 <b>Relationships:</b> Occlusion_1_3 hasOccluder Static_object2 Occlusion_1_3 hasOccluded PersonObj1
	<b>Image 3</b>	
<b>Previous knowledge</b>	TREN	<b>Instances:</b> Track_1 [position, size] <b>Relationships:</b> Track_1 isAssociatedTo Object PersonObj1
	SCOB	<b>Instances:</b> PersonObj_1, Static_object1, Static_object2, Static_object3, Static_object4 [positions and sizes] <b>Relationships:</b> PersonObj_1 hasAssociatedTrack Track_1
	ACTV	<b>Instances:</b> Occlusion_1_3 <b>Relationships:</b> Occlusion_1_3 hasOccluder Static_object2 Occlusion_1_3 hasOccluded PersonObj1

When the occlusion is fully detected some recommended behavior are sent directly to the tracking system prioritizing the knowledge system recommendations before the tracking algorithm detections. For the situation seen in 1B image the recommendations are re-create the track, maintain the last registered size before the deletion and situate the object centered in the object that occludes. The tracked object is automatically re-activated through the rule seen in Section 3. Table 2 shows a simplified view of how the specific data of the occlusion situation flow from low level to high level comprehension and scene interpretation.

## 5 Conclusion and Future Work

We have presented a general purpose approach based context knowledge and inference reasoning to deal with general and specific tracking problems at different fusion levels. An ontology based prototype has been developed to demonstrate the performance of this knowledge module. The system has been tested against the occlusion problem.

Future works will be addressed to expand the number of JDL model levels implemented. This extension will include ontologies to manage the impact of the recognized activities and the recommendations from the knowledge module to the tracking module. Therefore new rules, for abductive and deductive reasoning for these levels will have to be developed.

## Acknowledgement

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/ TIC-1485) and DPS2008-07029-C02-02.

## References

1. Gómez-Romero, J., Patricio, M.A., García, J., Molina, J.M.: Towards the implementation of an ontology-based reasoning system for visual information fusion. Applied Artificial Intelligence Group, Departament of Computer Science, Universidad Carlos III de Madrid (2009)
2. Gómez-Romero, J., Patricio, M.A., García, J., Molina, J.M.: Context-based reasoning using ontologies to adapt visual tracking in surveillance. In: Advanced Video and Signal Based Surveillance, pp. 226–231 (2009)
3. Gómez-Romero, J., Patricio, M.A., García, J., Molina, J.M.: Ontological representation of context knowledge for visual data fusion. In: 12<sup>th</sup> International Conference on Information Fusion, Ontological representation of context knowledge for visual data fusion, pp. 2136–2143 (2009)
4. Llinas, J., Bowman, C., Rogova, G., Steingber, A., Waltz, E., White, F.: Revisiting the JDL data fusion model II. In: 7<sup>th</sup> International Conference on Information Fusion, Stockholm, Sweden, pp. 1218–1230 (2004)

5. Randell, D.A., Cui, Z., Cohn, A.G.: A Spatial Logic based on Regions and Connection. In: Proceedings 3rd International Conference on Knowledge Representation and Reasoning (1992)
6. Sánchez, A.M., Patricio, M.A., García, J., Molina, J.M.: A Context Model and Reasoning System to Improve Object Tracking in Complex Scenarios. In: Expert Systems with Applications, vol. 36, pp. 10995–11005 (2009)
7. Serrano, M.A., Gracia, J., Patricio, M.A., Molina, J.M.: Interactive video annotation tool. In: Carvalho, A.P.L.F.D., Rodríguez-González, S., De Paz Santana, J.F., Rodríguez, J.M.C. (eds.) Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing, vol. 79, pp. 325–332. Springer, Heidelberg (2010)
8. Serrano, M.A., Patricio, M.A., García, J., Molina, J.M.: Dynamic RCC Automatically Updated for Topological Reasoning. In: Distributed Computing and Artificial Intelligence (2011) (accepted)
9. RACER. Renamed ABox and concept expression reasoner,  
<http://www.racer-systems.com/>
10. The video performance evaluation resource ground truth authoring tool (ViPER-GT),  
<http://viper-toolkit.sourceforge.net/products/gt/>