Context-based reasoning using ontologies to adapt visual tracking in surveillance

Juan Gómez-Romero, Miguel A. Patricio, Jesús García, José M. Molina
Applied Artificial Intelligence Group
Universidad Carlos III de Madrid
Colmenarejo, Spain
jgromero@inf.uc3m.es, mpatrici@inf.uc3m.es, jgherrer@inf.uc3m.es, molina@ia.uc3m.es

Abstract— **Classical tracking methods are often insufficient when dealing with complex scenarios. In order to solve tracking errors, innovative techniques based on the use of information about the context of the scene have been proposed. Context information ranges from precise measures computed on the pixels of the object neighborhood to high level representations of the entities and the activities of the scene. In this work, we focus on the second approach and propose an ontology-based extension of a general tracking procedure that reasons with abstract context descriptions to improve its accuracy. We describe the design of this extension and how reasoning is performed, as well as its advantages in surveillance scenarios.**

Keywords—object tracking; context; ontologies; automatic reasoning

I. INTRODUCTION

Tracking algorithms, mostly based on quantitative estimation methods, usually fail when dealing with complex scenarios. Complex scenarios present interactions between objects (both static and tracked, e.g. occlusions, unions, or separations), changes in the scene appearance (e.g. illumination), and modifications of the objects (e.g. deformations), which are difficult to manage and frequently result in tracking errors [1]: track discontinuity, inconsistent track labeling, inconsistent track size, etc. These problems are especially challenging in surveillance applications, since they require accurate identification of the entities of the scene and precise tracing of their movements.

Context knowledge has been proposed to be incorporated to computer vision systems in order to tackle complex scenario issues [2]. Context has been usually considered at a low abstraction level, in such a way that the context of an object is a numerical measure computed on the values of the pixels that are within its surroundings [3, 4]. This approximation to context exploitation is primarily quantitative, and aims at developing numerical procedures that implicitly take into account larger image sections and a priori knowledge.

Fewer approaches however have studied context from a more abstract perspective [5]. In the broadest sense, context can be considered to encompass all the additional information not directly provided by the visual sensors that can be used to understand what is happening in the scenario. From this point of view, context includes [6]: (i) information about the scene environment (structures, static objects, illumination and behavioral characteristics, etc.); (ii) information about the parameters of the recording (camera, image, and location features); (iii) information previously computed by the vision system (past detected events); (iv) user-requested information (data provided by human users). This approximation requires the use of symbolic knowledge formalisms, and aims at accomplishing cognitive interpretation of the scene as a whole by reasoning with explicit representations of perceptual and contextual data.

In this work, we study the advantages of this second approach to the use of context knowledge in object tracking. We investigate how to represent general context knowledge and how to reason with it to improve tracking processes. Context aids to interpret the situation and, in accordance with it, it can be applied to complete or rectify the tracking results, and attune the tracker. That is, context is used not only to recognize the scene, but also to provide feedback to the tracking algorithm in the form of suggestions or corrections. For instance, with a global description of the scene and the entities participating in it, it can be deduced that a group of people is moving together and performs a common action. Accordingly, the tracking algorithm should be recommended not to merge the individual tracks but to keep all of them in presence of occlusions and interactions.

We propose the use of ontologies encoded with the Ontology Web Language (OWL) [7] for context representation and reasoning. Ontologies are used to build abstract descriptions of the scene, in terms of symbolic entities. These descriptions are the input of the reasoning procedures, which detect or predict tracking errors incompatible with the current situation, and alert the tracking algorithm. Some advantages of using OWL ontologies are that they: (i) separate declarative and procedural knowledge, which facilitates decoupling track processing and context representation; (ii) support reasoning and deduction of new knowledge, since they are based on well-known Description Logics; (iii) promote reusability, extensibility, and standardization, which facilitates the reutilization of the models in diverse domains.

In this paper we present an ontology-based extension of a tracking system (such as the one described in [8]) that, based on perceptual and contextual information, supports scene recognition and improves tracking. The contextual layer receives the (quantitative) results obtained by the general tracking layer, processes this information in accordance with the context knowledge, and provides as an output a set of recommended actions to be performed by the tracking procedure. We describe the structure and the composition of the proposed model and how it proceeds to generate tracking recommendations. We illustrate the advantages of our

978-0-7695-3718-4/09 $25.00 © 2009 IEEE
DOI 10.1109/AVSS.2009.21

approach with an example of the use of the extended tracking system in surveillance. By creating appropriate domain-specific knowledge bases, the system architecture can be applied in other applications.

The remainder of the paper is organized as follows. In Sect. 2, we overview some related work pertaining to the use of context knowledge in computer vision and surveillance. In Sect. 3, we describe the architecture of the extended system. In Sect. 4, we introduce the ontologies that compose the context model and explain the transformation from quantitative to qualitative knowledge. We assume the reader to be familiar with the use of ontologies for knowledge representation. In Sect. 5, we clarify the details of the computation of tracking recommendations, paying special attention to the interaction between the tracking and the context layer. In Sect. 6, we exemplify the functioning of the system with a practical case on surveillance. Finally, the paper concludes with a brief discussion on the results and plans for future research work.

II. Related Work

The use of context knowledge to improve the cognitive capabilities of vision systems is a widely studied topic in computer science. Context has proved to be a crucial factor to recognize perceived scenes in different domains, and specifically in surveillance applications. In contrast to the predominant quantitative approaches, some early works proposed the creation of explicit knowledge representations to incorporate context information to the process of interpreting visual inputs [6, 9]. These works remark that the sources of context data are multiple: non-visual sensors, human inputs, measures on the environment, parameters of the capture, etc.

Most of the subsequent similar approximations have used ad hoc first order logic-based representation mechanisms (e.g. [10]) or specific-purpose models of the scene objects (e.g. [11]). Recently, ontologies have been acknowledged as suitable formalisms for representing context knowledge, especially from the data fusion perspective [12]. Ontologies promote interoperation between systems and knowledge reuse, which is essential in this area.

The research work in [13] presents an OWL ontology enhanced with rules to represent objects and actors in surveillance systems. Similarly, in [14] the authors depict a system for scene interpretation based on Description Logics and supported by the reasoning features of RACER (Renamed Abox Concept Expression Reasoner), an inference engine for OWL ontologies.

The scene interpretations (obtained by relying on formal context representations) can be used to refine less abstract image-processing procedures. Object identification problems have been solved by applying contextual information [15]. To the best of our knowledge, very few works have explored this possibility in tracking. The works in [5, 16] are preliminary approaches to the issues tackled in this paper.

III. System Architecture

The structural architecture of the context-based extension of the tracking system is depicted in Fig. 1. The schema shows the tracking system (the GTL, general tracking layer) and, built upon it, the context-based extension (the CL, context layer).

The GTL is a software program that executes the video chain to process raw images captured by a camera. The GTL usually encompasses various modules, which correspond to the successive stages of the tracking process: foreground detection, association, initialization / deletion, and trajectory generation. The GTL defines its particular programming-language data structures to represent image and track information.

The CL acts in cooperation with the GTL. The CL receives from the GTL tracking information, processes it, and provides as a result a set of recommendations or actions that should be performed by the GTL. Since the CL uses context knowledge to accomplish this objective, it additionally has context information as input, and scene interpretations as output. The key novelty of our approach is that the information in the CL is represented with a formal

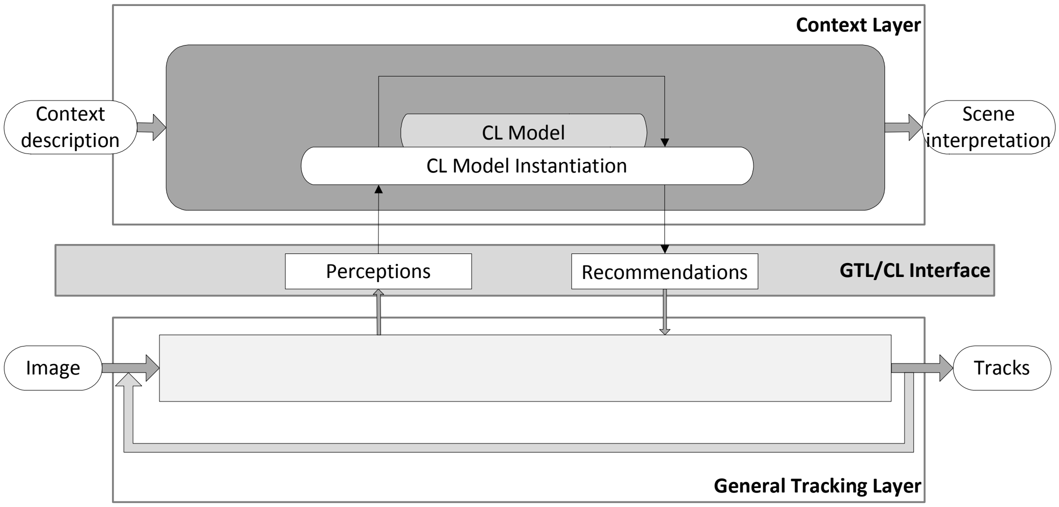

Figure 1. Structural architecture of the GTL and the CL.

knowledge model implemented as set of ontologies. The ontologies of the CL provide a terminology (concepts, relations, etc.) to describe scenes and context which is instantiated (with ontology individuals) in each execution with the data of a video sequence.

Fig. 2 depicts the functional architecture of the CL. On the left side, the schema shows the structure of the ontology-based representation model in various levels, from less abstract (track data) to more abstract (activity descriptions). Interpretation of acquired data (correspondence and recognition) can be seen as a transformation from knowledge expressed in a lower level ontology to knowledge expressed in a higher level ontology. Tracking data provided by the tracking system is transformed into ontology instances by accessing an intermediate interface, which updates the CL abstract scene model. More details of this knowledge representation and the transformations between levels are explained in Sect. 4.

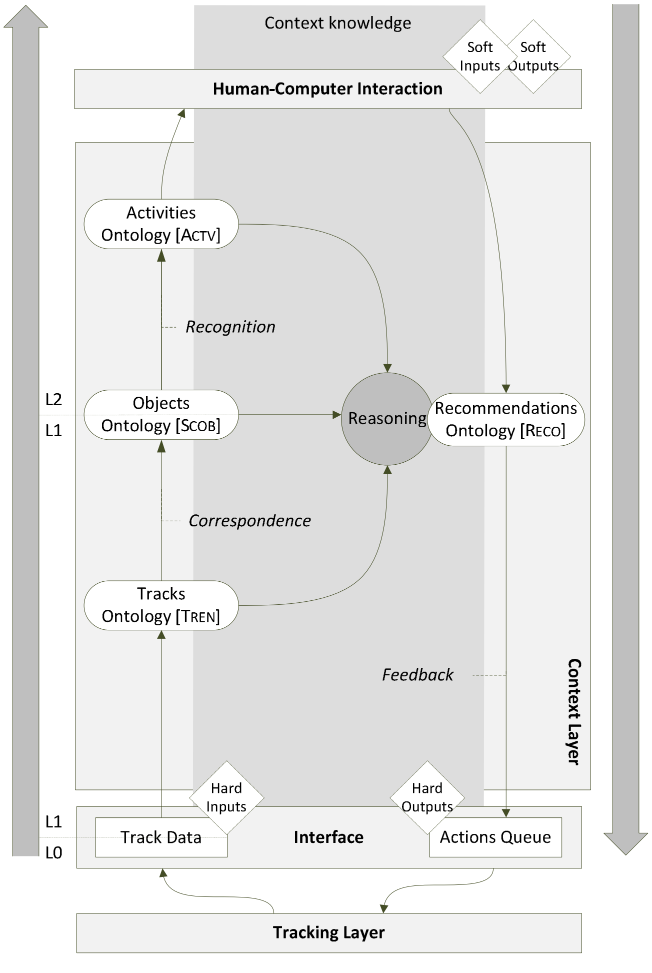

Figure 2. Functional architecture of the CL.

The eventual objective of the CL is to provide the GTL with appropriate recommendations that, according to the current scenario and context, can be used to correct and enhance its behavior. The calculation of these suggestions is performed in parallel to scene interpretation, since they can be obtained at different levels, as depicted on the right side of Fig. 2. This procedure is explained in Sect. 5.

The communication between the GTL and the CL is performed by means of an intermediate interface. The interface provides methods to access the model and to retrieve reasoning results. When tracks are created, modified, deleted, etc., the GTL invokes the input methods of the interface, and tracking data is transformed to the ontological representation. When CL reasoning processes are fired, resulting recommendations are placed in the actions queue, which is a data structure accessible to the tracking system. This interface allows maintaining independence between the GTL and the CL.

IV. ONTOLOGICAL REPRESENTATION AND ABDUCTIVE REASONING

A schema of the ontologies used to represent CL knowledge is depicted in Fig. 3. The modularization of the ontology has been designed in compliance to the JDL (Joint Directors of Laboratories) model for data fusion [17], specifically to the L1 (object assessment), L1½ (object relations) [18], and L2 (situation assessment) processing levels. We have separate ontologies to represent tracking data, scene objects, and activities, which are the main concepts of the model (marked in grey in the schema):

- *Tracking data* (JDL L1). The TREN (TRacking ENtities) ontology is a vocabulary to describe data from the tracking algorithm: tracks and track properties (color, position, velocity), frames, etc.
- *Scene objects* (JDL L1½). The SCOB (SCene OBjects) ontology is a vocabulary to describe real-world entities of the scene, properties, and relations: moving and static objects, topological relations, etc.
- *Activities* (JDL L2). The ACTV (ACTiVities) ontology is a vocabulary to describe behaviors: grouping, approaching, picking/leaving an object, etc.

We have developed a general version of these ontologies, in such a way that they can be specialized in each domain-specific application (Fig. 3). We provide a skeleton of the model that includes general concepts and relations. The developer must refine this vocabulary and extend the ontologies according with her objectives. For instance, the SCOB ontology defines a generic entrance object concept. In an indoor surveillance application, a door concept should be created by specialization of entrance object.

Another notable aspect of the ontological model is that we are interested in representing the temporal evolution of the scene, instead of its state in a given instant. That is, we want to keep all the information related to scene objects during the complete sequence, which changes between frames, and not only the lastly updated values. We have applied an ontology design pattern that solves this problem by creating an *observation* concept, which is related to the *observed* concept [19]. For example, in the TREN ontology, we have the Track and TrackSnapshot concepts. A Track instance is associated to several TrackSnapshot instances, each one of them encoding the values of the properties of a track (position, color, velocity, etc.) during some frames in which they do not change.

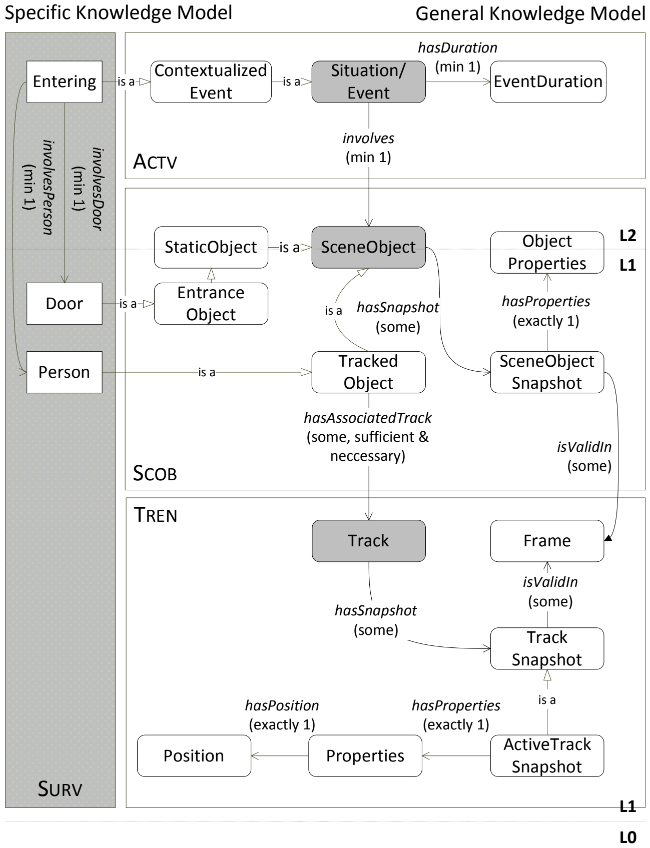

Figure 3. Excerpt of the CL ontology-based model (generic and specific)

Standard ontology reasoning procedures can be performed with the ontological representation of the CL knowledge: consistency tests, inclusion check, etc. Nevertheless, it does not state how the symbolic descriptions are built from the quantitative output of the tracking algorithm and the additional information. Therefore, it is necessary to incorporate tracking data into the ontological model, and to transform less abstract information to more abstract knowledge. In other words, mechanisms to reason between ontologies, and not only within ontologies, are required. The 'recognition' and 'correspondance' procedures depicted in Fig. 2 are examples of such reasoning tasks.

This procedure can be regarded as abductive reasoning, in contrast to the deductive reasoning performed within each ontology. Abductive reasoning takes a set of facts as the input and finds a suitable hypothesis that explains them. For instance, determining if a track (represented with an instance of the L1 ontology TREN) corresponds to a person or to a moving object (represented with instances of the L1½ ontology SCOB) is an example of this type of reasoning. Abductive reasoning is out of the scope of classical Description Logics [20], but in our case, it can be simulated by using customized procedures or, more interestingly, by defining transformation rules. Abduction rules can be created in a rule language and processed by an ontology-based reasoning engine, for instance the previously mentioned RACER. In Sect. 6, we show an example of the creation and the use of abduction rules.

V. REASONING FOR TRACKING IMPROVEMENT AND INTERACTION WITH THE GTL

The objective of the CL is not only to interpret activities with abductive procedures, but to enhance the results of the tracking algorithm. Once we have represented the observed scene with the CL ontological model, it is easy to define additional reasoning rules that, according to the high-level interpretation, detect tracking errors and suggest corrections to be considered by the tracking system, which only has a quantitative low-level perspective of the scene. Consquently, tracking-enhancement rules have a scene description in the antecedent and a recommendation specification in the consequent. Tracking-enhancement rules must be defined in a suitable language and processed by a reasoning engine (such as RACER). In Sect. 6, we present an example of tracking-enhancement rules.

The scene description of the antecedent of the tracking-enhancement rules is an expression built upon the terms defined in the ontologies of the CL. Hence, the concepts and the relations of the TREN, SCOB, and ACTV ontologies are used to create the *if* part of the rules, plus other specific predicates. The antecedent may be constructed at different abstraction levels. For example, at track level, a valid scene description is that a track is located in a determined position. At activity level, a valid scene description is that an object is about to leave the scene through a door. Terms at different abstraction levels could be even mixed in the same rule.

The recommended actions that participate in the consequent of the tracking-enhancement rules have been also defined with an ontology. The RECO (RECOmmendations) ontology is a vocabulary to describe suggestions to be the GTL. Recommendations are described at track level, i.e. they abstractly specify the action to be performed and the tracks that are implicated in this action. For example, a recommendation that suggests not creating a new track will be asserted to involve a track, and this track will be associated with property values. Instances of the RECO ontology are created as a result of CL calculations. The abstract RECO recommendations must be eventually converted to instructions that the GTL can execute.

An important element of the system architecture is the actions queue, which has been implemented to guarantee interoperability between the GTL and the CL. When one of the tracking-enhancement rules is fired, a recommendation is placed in the pending queue. Recommendations are marked with a time stamp and translated to concrete GTL actions by the interface. The incorporation of the actions queue is very convenient, since it decouples CL reasoning and GTL processing. The image-processing procedure does not need to be substantially changed, since communication is carried out by the interface methods, which make it transparent. The developer may even deactivate the context processing simply by changing a configuration option. The encapsulation of the recommendations queue also facilitates the development of conditional access mechanisms, e.g. priority ordering of the CL results, and asynchrony between the GTL and the CL.

VI. Example: CL in Surveillance

We have applied the ontological context-based approach to surveillance problems. The video sequence used in this example is part of the CLEAR2007 dataset[1]. In this recording, a person enters through a door (situated in the middle of the image) into an office room, which is the secured area. This dataset has been selected because we are interested in illustrating how the CL recognizes the activity of the scene, besides the correction of errors that may appear in the tracking process performed by the GTL.

We have developed the SURV (SURVeillance) ontology to be used as knowledge model in this application. This ontology imports the generic ontologies presented in Sect. 4 and Sect. 5, and specialize them with concepts such as Person, Door, and Entering (see Fig. 3). The ontology and the associated rules can be reused in other similar applications. The concrete values of the scene (e.g. position of person1) are created as instances of SURV. Before processing the video, the values of the static objects of the scene are introduced manually. In this case, we have marked three regions and three static objects: door1, table1, and screen1.

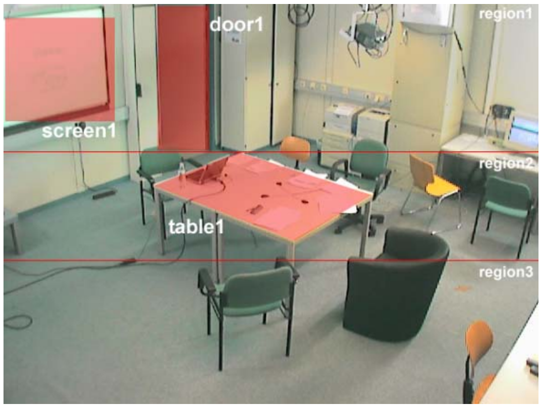

Figure 4. Markup of the static entities of the scene

We have also developed: (i) abductive rules to transform knowledge between ontology levels, as explained in Sect. 4; (ii) tracking-improvement rules to create recommendations, as explained in Sect. 5. An example of these rules is shown next. The rules are presented in the rule language accepted by the RACER reasoner. We assume that suitable implementation for new predicates (**width**, **height**, **inside**, **closeTo**), marked in bolds, have been also developed. Terms preceded by '?' are variables, and terms in italics are bounded values. Concept and property predicates are show in roman. The antecedent and the consequent of rules are interpreted as conjunctions.

Abductive rule AR1 (L1 to L1½). If a track is in region 1 and its size (w, h) is larger than (l1, l2), and it has not been identified with any other object, it corresponds to a person.

```
Track(?t) ^
TrackSnapshot(?tsn) ^
isAssociatedTo(?tsn, ?t) ^
isValidIn(?tsn, currentFrame) ^
not(hasAssociatedTrack(?old_o, ?t)) ^
inside(?tsn, region1) ^
width(?tsn, ?w) ^ height(?tsn, ?h) ^
greaterThan(?w, l1) ^
greaterThan(?h, l2)
-->
Person(?p) ^
hasAssociatedTrack(?p, ?t)
ObjectSnapshot(?new_psn) ^
hasSnapshot(?p,?new_psn) ^
isValidIn(?new_psn, currentFrame) ^
```

Abductive rule AR2 (L1½ to L2). If a new person in the scene is inside a door, the individual is entering the office.

```
Person(?p) ^
hasSnapshot(?p,?psn1) ^
isValidIn(?psn1,currentFrame) ^
not(
    hasSnapshot(?p,?psn2) ^
    isValidIn(?psn2, previousFrame)) ^
Door(?d) ^
closeTo(?psn1,?d)
-->
Entering(?act) ^
involvesPerson(?act, ?p) ^
involvesDoor(?act, ?d) ^
hasDuration(?act, ?duration) ^
begins(?duration, ?currentFrame)
```

Tracking enhacement rule TR1. If a person is entering the scene, notify the GTL that the associated track is completely new.

```
Entering(?act) ^
involvesPerson(?act, ?p) ^
hasAssociatedTrack(?p, ?t) ^
-->
NewTrackRecommendation(?rec) ^
affect(?rec, ?t) ^
```

When the person enters the scene (Fig. 5), the GTL detects a new moving entity and invokes the interface. The interface transforms the numerical values to ontology individuals and updates the current instantiation of the model of the scene. Then, reasoning processes are triggered and new instances of the model are created. The processing is the following:

1. track1 individual is created.
2. track1 and its associated values match AR1, which fires and creates person1, a new instance of Person that has associated track1.

[1] CLEAR (Classification of Events, Activities and Relationships) dataset was deployed to evaluate systems that are designed to recognize events, activities, and their relationships in interaction scenarios [21].

3. person1 values match AR2, which fires and creates act, a new instance of Entering that has associated person1 and door1.
4. act values match TR1, which fires and creates rec, a new instance of NewTrackRecommendation that has associated t, the track related to person1.

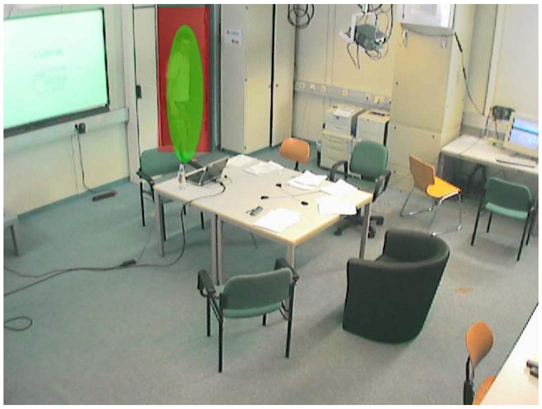

Figure 5. New moving entity is detected.

As the final result of the first call to the interface, a new recommendation is created and inserted in the queue. The GTL will subsequently consult the queue by using a suitable interface method in order to fetch pending recommendations and to act in consequence.

VII. Conclusions and Future Work

In this paper, we have presented an extension of a tracking system that uses ontologies to represent and reason with context knowledge in order to avoid issues arisen in complex scenarios. Ontologies are applied to build a knowledge model that supports recognition of scenes and tracking improvement. The architecture is extensible to other application, which will require the development of suitable representation models. To reduce this effort, we have provided a set of reference ontologies to be extended in each case. The model is readable, which additionally facilitates the incorporation of soft entries to the system. Particularly, the approach can be applied to surveillance applications.

We plan to continue this research work by fully integrating the context layer with the tracking software. This will probably require adapting the ontological model and the procedures of the CL, which may be too resource-consuming for computer vision. The implementation will be extensively tested with existing datasets to demonstrate beyond the presented example that the contextual layer effectively reduces tracking errors, and to quantify the improvement with respect to other methods.

Acknowledgment

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, SINPROB, CAM MADRINET S-0505/TIC/0255 and DPS2008-07029-C02-02.

References

[1] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” ACM Computing Surveys, 38(4): no 13, 2006.

[2] B. Draper, A. Hanson, and E. Riseman, “Knowledge-directed vision: Control, learning, and integration,” Proceedings of the IEEE, 84(11):1625-1637, 1996.

[3] A. Torralba, “Contextual Priming for Object Detection,”. International Journal of Computer Vision, 53(2): 169-191, 2003.

[4] J. Wang, P. Neskovic, L. N. Cooper, “Context-based tracking of object features,” Proc. IEEE Int. Joint Conference on Neural Networks (IJCNN 2004), Budapest, Hungary, 2004, pp. 1775-1779.

[5] A. Sánchez, M.A. Patricio, J. García, and J.M. Molina, “A context model and reasoning system to improve object tracking in complex scenarios,” Expert Systems with Applications, 36(8): 10995-11005, 2009.

[6] F. Bremond, and M. Thonnat., “A context representation for surveillance systems,” Proc. ECCV Workshop on Conceptual Descriptions from Images, Cambridge, UK, 1996.

[7] I. Horrocks, “Ontologies and the Semantic Web,” Communications of the ACM 51(12): 58-67, 2008.

[8] M.A. Patricio, F. Castanedo, A. Berlanga, O. Pérez, J. García, and J.M. Molina, “Computational intelligence in visual sensor networks: Improving video processing systems,” Computational Intelligence in Multimedia Processing: Recent Advances, Springer, 2008, pp. 351-377.

[9] T.M. Strat, “Employing contextual information in computer vision,” Proc. ARPA Image Understanding Workshop, Washington, USA, 1993, pp. 217-229.

[10] O. Brdiczka, P.C. Yuen, S. Zaidenberg, P. Reignier, and J.L. Crowley, “Automatic acquisition of context models and its application to video surveillance,” Proc. 18th Int. Conf. on Pattern Recognition, Hong Kong, 2006, pp. 1175-1178.

[11] Y. Huang, T. S. Huang, “Model-based human body tracking,” Proc. 16th Int. Conference onf Pattern Recognition (ICPR’02), Quebec, Canada, 2002, pp. 552, 555.

[12] A. Steinberg, and G. Rogova, “Situation and context in data fusion and natural language understanding,” Proc. 11th Int. Conf. on Information Fusion, Cologne, Germany, 2008, pp. 1-8.

[13] L. Snidaro, M. Belluz, and G.L. Foresti, “Domain knowledge for surveillance applications,” Proc. 10th Int. Conf. on Information Fusion, Quebec, Canada, 2007, pp. 1-6.

[14] B. Neumann, and R. Möller, “On scene interpretation with Description Logics,” Image and Vision Computing, 26: 82-101, 2008.

[15] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V. Papastathis, and M. Strintzis, “Knowledge-assisted semantic video object detection,” IEEE Transactions on Circuits and Systems for Video Technology 15(10): 1210-1224, 2005.

[16] A. M. Sánchez, M.A. Patricio, J. García, J. M. Molina, “Video Tracking Improvement Using Context-Based Information,” Proc. 10th Int. Conference on Information Fusion, Quebec, Canada, 2007, pp. 1-7.

[17] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White, “Revisiting the JDL data fusion model II,” Proc. 7th Int. Conf. on Information Fusion, Stockholm, Sweden, 2004, pp. 1218-1230.

[18] S. Das, “High-Level Data Fusion,” Artech House Pub., 2008.

[19] N. Noy, and A. Rector, “Defining n-ary relations on the Semantic Web,” 2006. (http://www.w3.org/TR/swbp-n-aryRelations/).

[20] C. Elsenbroich, O. Kutz, and U. Sattler, “A case for abductive reasoning over ontologies,” Proc. OWL: Experiences and Directions Workshop, Athens, Georgia, USA, 2006.

[21] R. Stiefelhagen, K. Bernardin, R. Bowers, R.T. Rose, M. Michel, J. Garofolo, “The CLEAR 2007 Evaluation”, Proc. Multimodal Technologies For Perception of Humans, Baltimore, MD, USA, 2007, pp. 3-34.