Communication in distributed tracking systems: an ontology-based approach to improve cooperation

Juan Gómez-Romero*, Miguel Á. Patricio, Jesús García and José M. Molina
Applied Artificial Intelligence Group, University Carlos III os Madrid, Spain
*Email: jgromero@inf.uc3m.es

Abstract: *Current Computer Vision systems are expected to allow for the management of data acquired by physically distributed cameras. This is especially the case for modern surveillance systems, which require communication between components and a combination of their outputs in order to obtain a complete view of the scene. Information fusion techniques have been successfully applied in this area, but several problems remain unsolved. One of them is the increasing need for coordination and cooperation between independent and heterogeneous cameras. A solution to achieve an understanding between them is to use a common and well-defined message content vocabulary. In this research work, we present a formal ontology aimed at the symbolic representation of visual data, mainly detected tracks corresponding to real-world moving objects. Such an ontological representation provides support for spontaneous communication and component interoperability, increases system scalability and facilitates the development of high-level fusion procedures. The ontology is used by the agents of Cooperative Surveillance Multi-Agent System, our multi-agent framework for multi-camera surveillance systems.*

Keywords: information fusion, tracking, video surveillance, ontologies, multi-agent systems

1. Introduction

In recent years, the interest in Computer Vision systems has reached new heights. In the simplest case, vision systems have a single sensor, which provides a sequence of frames to the image processing algorithms. These algorithms, mainly based on statistical prediction and inference methods, aim to automatically detect and trace the entities in the observation area, and to recognize and predict the actions that they are performing to act consequently. In the most complex case, systems encompass several distributed sensors, which additionally requires to fuse information acquired at different locations of the sensor network. Among Computer Vision tasks, tracking – the estimation of the number of objects in a scene, together with their instantaneous locations, kinematic states and any other characteristics required – is one of the most studied, and a first step before performing more intricate video analysis procedures (Yilmaz *et al.*, 2006).

Surveillance is a typical application domain where accurate tracking is required. Recently, the decreasing price of video camera hardware and the development of network technologies have given rise to the development of third-generation surveillance systems (Regazzoni *et al.*, 2001; Valera & Velastin, 2005). This term designates systems that resemble the nature of the human intelligent process of surveillance (which activates certain cognitive abilities) and satisfy the requirements of modern applications (large number of cameras, geographical spread of resources, many monitoring points, etc.) Two difficulties can be identified in such distributed systems. Firstly, it is necessary to implement suitable procedures to integrate data generated at each location, in order to obtain a high-level interpretation of the whole scene. Secondly, the scalability of the system must be guaranteed, which can be difficult when new heterogeneous cameras are incorporated to build a large and scattered circuit.

In previous researches, we presented the Cooperative Surveillance Multi-Agent System (CS-MAS), an agent-based model and platform to support the development of third-generation surveillance systems (Patricio *et al.*, 2007; Castanedo *et al.*, 2010). CS-MAS proposes the formation of smart camera coalitions; i.e., groups of sensors able to carry out complex processing tasks and cooperate with their neighbours by means of sophisticated interaction protocols. The combination of information acquired by geographically distributed cameras in CS-MAS improves tracking results in surveillance applications, since multiple fields of view are considered. Additionally, distribution increases system robustness and fault tolerance, since the same information may be captured and replicated at different points of the network.

Nevertheless, CS-MAS does not completely solve two difficulties. First, it is necessary to implement suitable procedures to represent data generated at each location to obtain a high-level and global interpretation of the scene. Second, the scalability of the system must be guaranteed, which can be difficult when new heterogeneous cameras are incorporated to build a large and scattered circuit.

To overcome these problems, we proposed a semantic model to represent the visual information shared in distributed artificial vision systems (Gómez-Romero *et al.*, 2009c). In the current paper, we extend that preliminar work to provide a detailed description of the ontology to represent the tracking information managed by agents and interchanged in CS-MAS. We also explain how the ontological representation is integrated within the CS-MAS architecture and how concepts and relations are used to express and communicate agents' beliefs. As a major contribution, we show that the ontology behaves as an agreed vocabulary that allows tracking data to be represented in a symbolic, common and understandable way. Thus, information exchange is decoupled from information processing, which facilitates the incorporation of extended, heterogeneous and/or third-party components into the vision system. The advantages of this approach can be exploited in scenarios requiring complex agent interactions, such as camera handover. Interestingly enough, further functionalities to reason with high-level abstractions can be implemented on top of the ontological representation.

Ontologies are a state-of-the-art knowledge representation and reasoning formalism. They have proved to be valid in several scenarios that require interoperation between heterogeneous entities, e.g. the Semantic Web (Horrocks, 2008). Ontologies have strong underpinnings in Description Logics (DLs) (Baader *et al.*, 2003, 2008); in fact, ontology languages are usually (equivalent to) a decidable DL, as in the case of the standard OWL (Horrocks & Patel-Schneider, 2004) and its successor OWL 2 (Hitzler *et al.*, 2008) – used in this work. Reasoning with ontologies is an automatic procedure that infers new axioms that have not been explicitly included in the knowledge base but are logical consequences of the represented axioms. An advantage of ontologies over classical multi-agent content languages, such as FIPA Semantic Language, is that the latter are undecidable in their general form – i.e., it is not guaranteed that all the inferences are computable in a finite time, whereas the former are decidable and are supported by current tools (APIs, reasoners, etc.) Hence, ontologies have been proposed to be the knowledge representation of agent systems (Hendler, 2001; Schiemann & Schreiber, 2006; Erdur & Seylan, 2008).

The rest of this paper is organized as follows: next, we describe the context of our research and review related work on the use of ontologies in (distributed) Computer Vision systems. In section 3, we explain the role of the ontological representation inside the CS-MAS framework. In section 4, we describe the formulation of the ontology, whereas in section 5, we illustrate its use along with CS-MAS in a surveillance scenario, including instance creation, message-passing and support for high-level information fusion. Finally, the paper concludes with a discussion of our proposal and plans for future research work.

2. Related work

Data integration and interpretation problems in Computer Vision have been faced by applying data and information fusion techniques. Fusion techniques combine data from multiple sensors and related information to achieve more specific inferences than could be achieved by using a single, independent sensor (Hall & Llinas, 2009). Data fusion systems are usually organized by following the guidelines of the Joint Directors of Laboratories (JDL) model (Steinberg & Bowman, 2009). JDL classifies fusion processes according to the abstraction and the refinement of the entities involved. The canonical JDL model establishes five operational levels in the transformation of input signals to decision-ready knowledge (Llinas *et al*., 2004; Steinberg & Bowman, 2004), namely: signal feature assessment (L0); entity assessment (L1); situation assessment (L2); impact assessment (L3) and process assessment (L4). Tracking is considered an L1 task, since it aims at estimating the properties of isolated objects from pre-processed sensor signals (L0), instead of the relations between them. High-level information fusion, corresponding to the levels L2 and L3 of the JDL model, aims at obtaining a description of the relations between the objects in the perceived scenario. These relations are usually expressed with interpretable symbolic terms (e.g., actions, intentions, threats), instead of the usual numerical measures (e.g., density functions, movement vectors) calculated in L1. L4 tasks are aimed at planning and performing procedures to improve the whole fusion process, from low-level data acquisition to high-level situation assessment.

Conceptual models to acquire, represent and exploit formal knowledge in fusion have been extensively proposed (Nowak, 2003). Ontologies have gained popularity in the last few years and are being applied to solve fusion problems in Computer Vision systems. Recent researches can be classified according to the levels defined by the JDL model.

At image-data level (i.e. JDL level 0), one of the most important contributions is Core Ontology for MultiMedia (COMM), an OWL ontology to encode MPEG-7 data (Arndt *et al*., 2008). COMM does not represent high-level entities of the scene, such as tracks, objects or events. Instead, it identifies the components of an MPEG-7 video sequence in order to link them to (Semantic) Web resources. Similarly, the Media Annotations Working Group of the W3C is working in an OWL-based language for adding metadata to Web images and videos (Lee *et al*., 2009). These approaches do not specifically aim at representing scene data, but the structure of the acquired video sequence, focusing on the normalization of the several existing video formats.

More related to our approach are such proposals targeted at modelling video content at the object level (i.e. JDL L1). For example, François *et al*. (2005) have created a framework for video event representation and annotation. In this framework, Video Event Representation Language (VERL) defines the concepts to describe processes (entities, events, time, composition operations, etc.), and Video Event Markup Language (VEML) is an XML-based vocabulary to markup video sequences (scenes, samples, streams, etc.). VEML 2.0 has been expressed in OWL, but only partially because it imports VERL elements that need a more expressive language. Moreover, the limitation in the number of entities represented in VEML 2.0 reduces its usefulness, as it is discussed by Westermann and Jain (2007), who present a framework that supports representation of uncertain knowledge. An approach that stands halfway between data and object level is due to Kokar and Wang (2002). In this research work, the data managed by the tracking algorithm are represented symbolically in a similar fashion as we do, but they do not take into account the particularities of using the vocabulary for information transmission. Similarly, Snidaro *et al*. 2007) have presented a first approach to the development of a tracking data ontology, a work that is still in progress.

At a more abstract level (i.e. JDL L2 and L3), scene interpretation issues are being dealt with ontologies as well. In Snidaro and Foresti (2007), the authors discuss the use of OWL ontologies and SWRL rules to define and recognize scene events in Ambient Intelligence domains. In Neumann and Möller (2008), an ad-hoc proposal for scene interpretation based on DLs is presented. The paper shows how the reasoning features of the RACER reasoning engine provide functionalities that support scene recognition. The approach is hardly generalizable, but illustrates the expressivity of DLs for such tasks and the existence of appropriate tools. The problem of representing high-level semantics of situations with a computable formalism is also faced in Kokar *et al.* (2009). The authors present an OWL ontology (Situation Theory Ontology, STO) that encodes Barwise's situation semantics. Both research works tackle the problem of transforming numeric data into symbolic objects, because scene interpretation must eventually take raw video data as an input. Our representation has been purposely designed to solve this problem and could be used in combination with these high-level approaches.

In this regard, in previous research works, we have developed a framework for contextual scene recognition in surveillance applications and tracking enhancement (Gómez-Romero *et al.*, 2009a). More abstract objects (e.g. people, moving items) with special features or behaviours have been defined by relying on the ontology hereby presented (Gómez-Romero *et al.*, 2009b), in addition to scene interpretation rules (Gómez-Romero *et al.*, 2009d). Thus, the ontology presented in the current paper is complementary to these approaches. While they are especially focused on high-level interpretation in one-camera configurations, the ontology described in this work may be used to bridge the gap between real-world physical images and high-level symbolic interpretation, which is known as the grounding problem (Pinz *et al.*, 2008).

3. Ontological knowledge representation in CS-MAS

CS-MAS is a multi-agent framework for visual sensor networks especially adapted to surveillance environments. The framework provides a reference platform to organize, communicate and coordinate all the procedures carried out by a distributed vision system, focusing on data fusion for tracking. CS-MAS uses deliberative processes to conduct the fusion of information between neighbour cameras and to manage coordinated decision-making in the sensor network. Essentially, each camera is represented and managed by an individual deliberative and social software agent – a Belief Desire Intention (BDI) agent (Rao & Georgeff, 1995) (Figure 1).

In principle, agents can only know the part of the scenario in their field of view. In order to avoid errors due to local knowledge of the world, agents establish social relations to interchange visual information and increase their knowledge, in such a way that they get a better picture of the scenario and make better decisions. Thus, the content of the agents' messages is tracking data, which is represented with an ontology. In this section, we explain how agents acquire and manage tracking data (i.e., the video processing role, in which the ontology is used to represent agents' beliefs), and how tracking data are communicated to other agents (i.e., the communication role, in which the ontology is used as the content language). In the next section, we describe in more detail the classes, relations and axioms that compose the tracking data ontology.CS-MAS has been built on the JADEX platform (Braubach *et al.*, 2005), a Java-based framework for fast development of multi-agent systems based on the Java Agent Development Environment (JADE) platform (Bellifemine *et al.*, 2007). Management of OWL messages is supported by the Pellet inference engine and the OWL API interface (Horridge *et al.*, 2007).

3.1. Video processing

Processing of camera data is performed by agents in CS-MAS at two logical levels: the tracking layer and the BDI layer. First, each camera is associated with a process that acquires current estimates. This process is mainly based on a tracking subsystem, which sequentially executes various image-processing algorithms that detect and trace all the targets within the local field of view. This *tracking layer* is arranged in a pipelined structure

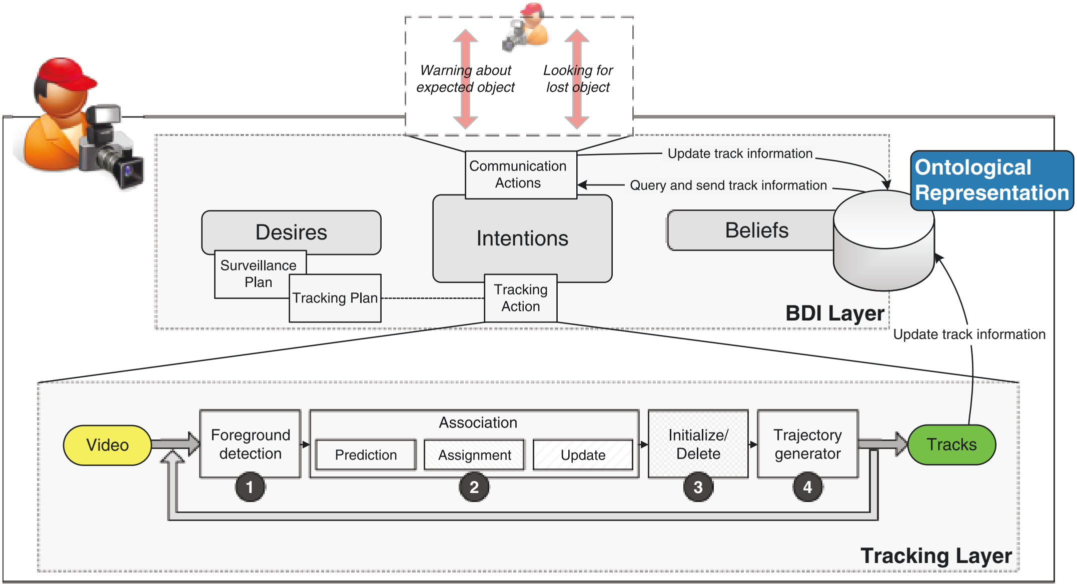


Figure 1: *CS-MAS BDI agent architecture.*

of several modules, as shown in Figure 1, which correspond to the successive stages of the tracking process (Besada *et al*., 2005): (1) movement detection (background and foreground detection); (2) blob[1]-track[2] association, which includes prediction, assignment and update; (3) track creation and deletion; and (4) trajectory generation.

The *BDI layer* uses the ontology to encode these perceptions acquired by the agent. At this level, the purpose of the ontology is to serve as a symbolic representation of the tracking numerical estimates. The use of an OWL ontology facilitates the manipulation of data and supports the first step in the scene interpretation procedure. As mentioned, the framework applies the BDI paradigm to model agents. The beliefs of the agents are represented as instances of the ontology, whereas desires and intentions are included in JADEX format. In our domain, we suggest that the beliefs, desires and intentions of each camera-agent are the following.

3.1.1. Beliefs Agent beliefs will include information about the outside world, like objects that are being tracked – location, size, trajectory, etc. – contextual information – entities that might require special attention – and geographic information about the camera itself – location, neighbour cameras, etc. The belief base of the agent will be updated with the new perceived information. It may also be convenient to constrain the stored beliefs in a temporal window, in order to avoid the overhead of keeping all past knowledge. Therefore, the ontology must include convenient classes to describe tracks and track properties changing in time. The classes defined in the ontology are explained in section 4, whereas an example of instances is presented in section 5.1.

3.1.2. Desires Since the final goal of agents is the correct tracking of moving objects, they have two main aims: permanent surveillance and temporary tracking. The surveillance plan is continuously executed by agents. Camera-agents permanently capture images from the camera until an intruder is detected or announced by a warning from another agent; in these cases, the tracking plan is fired. The tracking plan runs

[1] *A blob is a set of pixels that form a connected region.*

[2] *A track is a low-level representation of a moving entity. It is represented as a single blob or as a set of related blobs with properties: size, colour, velocity, etc.*

internally a tracking process (implemented at the tracking layer) on the images from the camera until it is no longer possible. The tracking plan includes suitable actions to update the beliefs of the agent; that is, to provide the track estimations to the BDI layer.

3.1.3. Intentions Agents perform two types of actions: internal and external. Internal actions are related to video processing and tracking, and involve the issue of commands to the tracking subsystem or the camera. External actions correspond to communication acts with other agents. The agents will send and receive messages packaging beliefs, which are represented with the ontology. The protocols and the messages used to communicate agent information are described in the next subsection.

3.2. Agent communication

Agent communication in CS-MAS is performed by interchanging FIPA-compliant messages[3], the standard for communication in multi-agent systems. The use of standard FIPA messages with contents represented with our ontology promotes interoperability in the platform, as well as the incorporation of new heterogeneous agents. Two main types of dialogues can occur between agents in the framework.

3.2.1. Warning about expected object dialogue This dialogue warns neighbour agents about the expected presence of a moving object in their field of view. An agent receiving this message may acknowledge the warning by returning a confirmation to the sending agent. The goal is to compel near agents to initialize plans for tracking a moving object, once the sending agent realizes that some circumstances in the very near future will make it lose the track (see next section).

Since more than one neighbour agent may be interested in the advice, the warning is sent with the FIPA performative *call for proposals*. This message contains the estimated features of the tracked object. Agents may use global positioning or references to shared contextual objects (e.g., doors or windows) to describe an estimated target. We show in Figure 2(a) the structure of this message, where `?i` and `?j` are agent identifiers and track description is an ontological description of the properties of the track.

The answer to this call is a *propose* message in which the warned agent informs the warning agent about its intention of performing the action implicitly suggested (Figure 2(b)). The answer may include a complex description to inform about the conditions that the scenario must satisfy before beginning the requested action. This condition is expressed in the common ontological vocabulary.

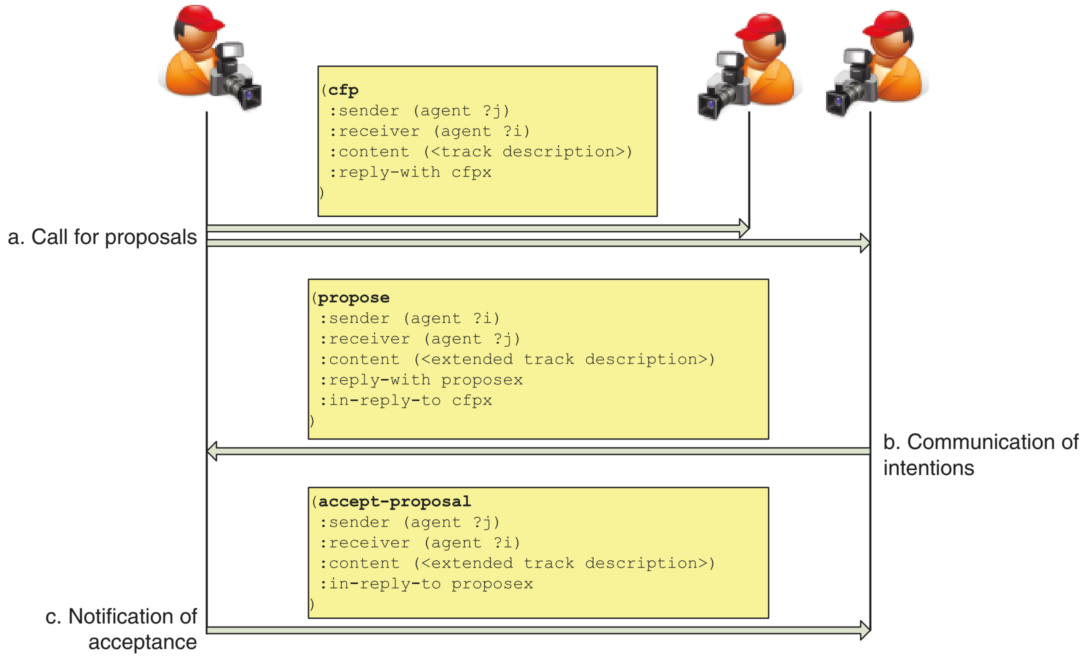
Finally, the agent who initially made the call may send a notification accepting the proposal of the neighbour agent (Figure 2(c)). The agreement is communicated with an *accept-proposal* message expressing conformance with the delegation of the task.

3.2.2. Looking for lost object dialogue This dialogue takes place when a track suddenly disappears from the field of view of an agent, e.g. due to an occlusion. With this dialogue, an agent polls neighbour agents to discover whether any of them is detecting the missing object.The initial message of the dialogue is a query to camera agents that are potential observers of the moving objects (Figure 3(a)). The FIPA performative is *query-if*, which is used to ask whether another agent believes that a given proposition is true. In this case, the proposition is a description of the missing track. The answer to this question is an *inform* message with the requested information, if available, plus any additional fact that the answering agent may consider interesting (Figure 3(b)).

4. The tracking entities description ontology

A DL ontology is developed from the following elements: concept or classes, which represent the basic ideas of the domain; instances or individuals, which are concrete occurrences of concepts; and relations, roles or properties, which represent binary connections between individuals or individuals and typed values. Complex

[3] *http://www.fipa.org*

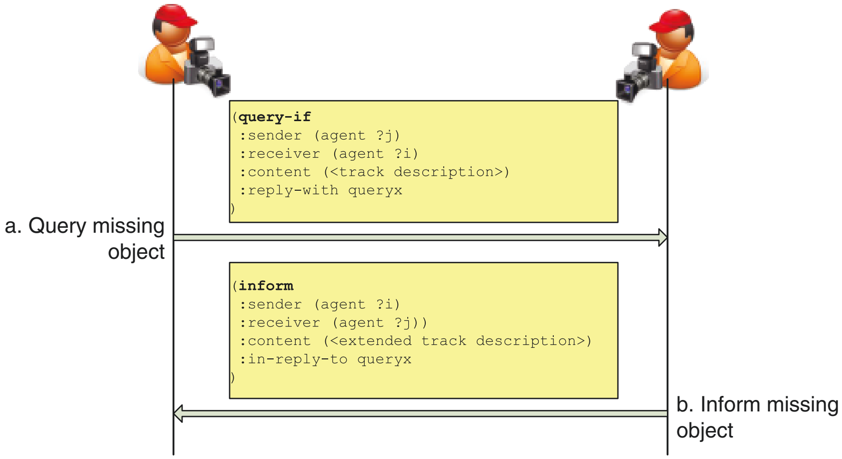


```
(cfp
 :sender (agent ?j)
 :receiver (agent ?i)
 :content (<track description>)
 :reply-with cfpx
)
```

```
(propose
 :sender (agent ?i)
 :receiver (agent ?j)
 :content (<extended track description>)
 :reply-with proposex
 :in-reply-to cfpx
)
```

```
(accept-proposal
 :sender (agent ?j)
 :receiver (agent ?i)
 :content (<extended track description>)
 :in-reply-to proposex
)
```

Figure 2: Warning about expected object *dialogue*.



```
(query-if
 :sender (agent ?j)
 :receiver (agent ?i)
 :content (<track description>)
 :reply-with queryx
)
```

```
(inform
 :sender (agent ?i)
 :receiver (agent ?j))
 :content (<extended track description>)
 :in-reply-to queryx
)
```

Figure 3: Looking for lost object *dialogue*.

concept and relation expressions can be derived inductively from atomic primitives by applying the constructors defined by the logic. Domain knowledge is represented by asserting axioms, which establish restrictions over concepts, instances and relations, describing their attributes by delimiting their possible realization. A DL ontology is a triple encompassing a TBox and a RBox, which contain terminological axioms (respectively, axioms about concepts and roles), and an ABox, which contains extensional axioms (axioms about individuals).

The Tracking Entities Description Ontology (TREN) defines a vocabulary to describe visual information. That is, the ontology includes a set of concepts, relations and axioms to describe tracking data that are shared for all the agents. This vocabulary supports belief management within each agent and interchange of tracking information, according to the previously mentioned messages. Agents manage a local instantiation of the ontology, where ontology individuals corresponding to runtime scenario data are created. Therefore, all the agents use the same terms (defined in the TBox and the RBox of the global ontology) to describe the perceptions on their field of view (created as instances of their local ABox). In this section, we explain the development of the terminological part of the ontology, whereas in section 5, we present an example of the creation of ontology instances.

Figure 4 presents a simplified schema of the overall structure of the TREN ontology. Concepts are depicted with squared boxes, whereas axioms are depicted with links between concepts.

In order to separate the three dimensions of the knowledge represented in TREN, we have defined three abstract concepts: Geometric Concept, TemporalConcept and CVConcept. Concepts aimed at describing the geometric and appearance aspects of a tracked entity – e.g., size, position, colour – inherit from GeometricConcept. Similarly, the temporal aspects of a tracked entity – e.g., capture time – are represented with concepts descending from TemporalConcept The core concepts of TREN used to describe the tracked entities – Computer Vision concepts: e.g., Frame, Track, TrackSnapshots, ActualProperties – are descendants of the CVConcept.

A TREN Frame is identified by a numerical ID and can be marked with a time stamp. The ontology imports the OWL-Time ontology to associate a DateTimeDescription to each frame instance (Hobbs & Pan, 2006). DateTimeDescription allows the representation of a time period in different scales (year, week, hour, minute, etc.). Frames may be related with the image as it has been captured by the camera, which must be an individual of RawData. Actually, RawData does not stores the raw frame itself, but a link to a file that should be resolved by the agent.

A Track is a moving entity of the scene. Tracks are labelled with an ID number. The representation of tracks is more complex than the representation of frames, since it is necessary to store their temporal evolution in order to reflect track continuity. We want to keep all the information related to a track during a complete sequence (activity, occlusions, position, size, velocity, etc.), which changes between frames, and not only its lastly updated values. Accordingly, we have defined a representation schema to describe track states and properties that evolve in time.

On the one hand, we associate to each track some property values that are valid only during some frames. To solve this issue, we have followed an ontology design pattern proposed by the W3C Semantic Web Best Practices and Deployment Working Group to define ternary relations in OWL ontologies (Noy & Rector, 2006). The ontology provides the hasSnapshot property to connect a set of TrackSnapshots to each Track. Each TrackSnapshot has constant property values that are asserted to be valid in various frames with the properties isValidInBegin and isValidInEnd. Additionally, the ontology defines different types of TrackSnapshot to distinguish between the basic states of a track: active (ActiveTrackSnapshot), grouped (GroupedTrackSnapshot) and occluded (OccludedTrackSnapshot). It can be seen that other states can be easily added to the representation.

On the other hand, track features must be defined as general as possible, in such a way that they can be extended. To solve this issue, we have followed the *qualia* approach, used in the upper ontology DOLCE (Gangemi *et al.*, 2002). This knowledge representation pattern distinguishes between properties themselves and the space in which they take values. This way, we have track qualities reified as concepts, such as Position or Size. In the case of Position, it is related to the property positionValue to a single instance of the concept PositionValue. A 2DPoint is a possible PositionValue. The remaining properties are defined similarly: Colour, Area, Size, etc.

The definition of geometrical concepts has been developed according to the work by Maillot *et al.* (2004), who propose primitive concepts such as Point, PointSet, Curve (a subclass of PointSet) or Polygon (a kind of Curve). It is interesting to highlight that properties are only associated to ActiveTrackSnapshots, in such a way that they are related to one ActualProperties set of detected

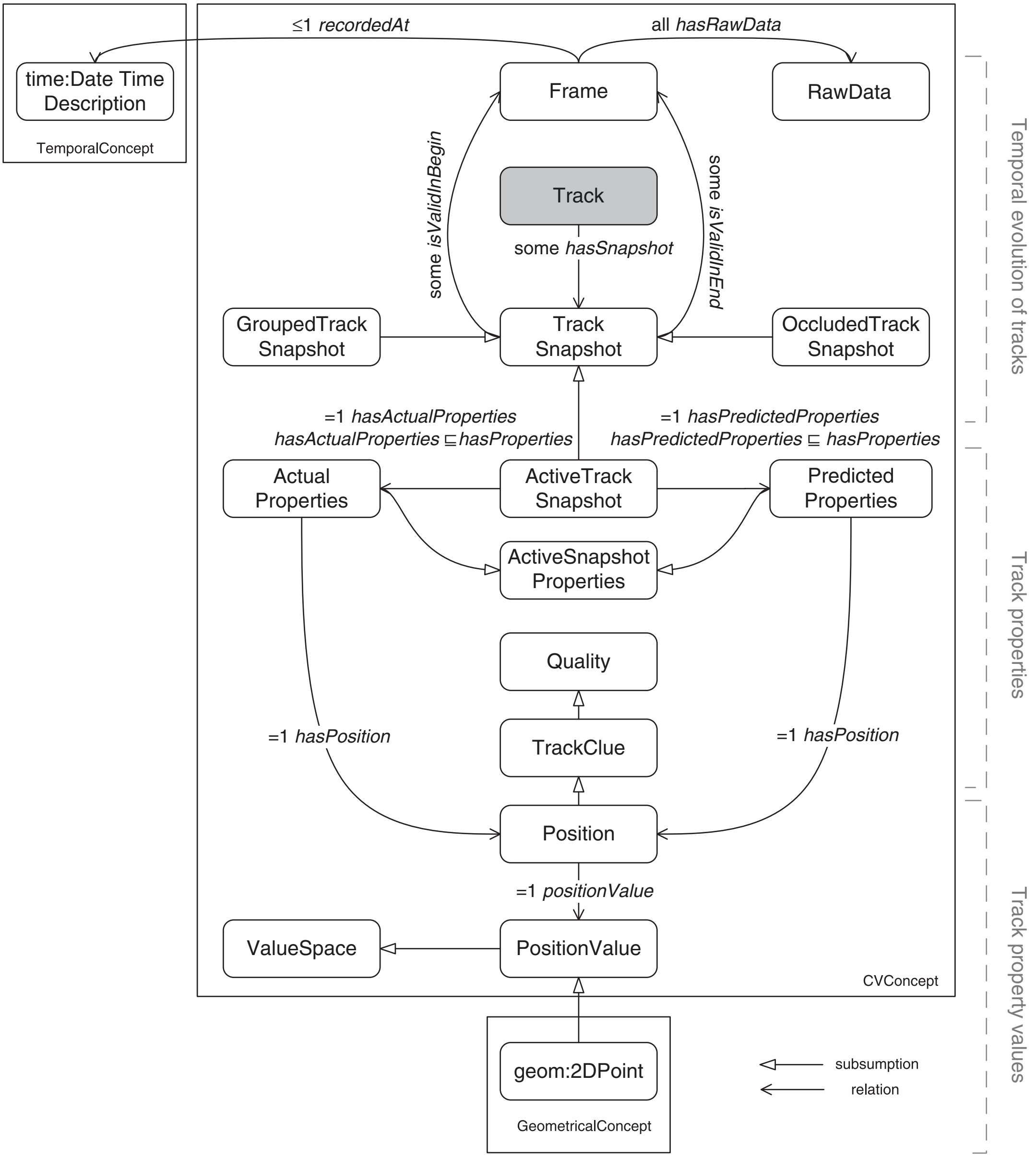


Figure 4: *Excerpt of the* TREN *Ontology.*

property values and to one PredictedProperties set of estimated property values.

Instances of TREN are created as a result of the processing of the video sequence performed by camera agents. The agent transforms the numerical tracking data provided by the image-processing algorithm into corresponding TREN instances. After this, the symbolic knowledge about the scene is ready to be delivered to other agents. This process is shown in the next section.

The ontology has been developed with the Protégé editor[4], version 4, which supports the extensions of the OWL 2 specification we have used. The ontology is publicly available at the

[4] *http://protege.stanford.edu*

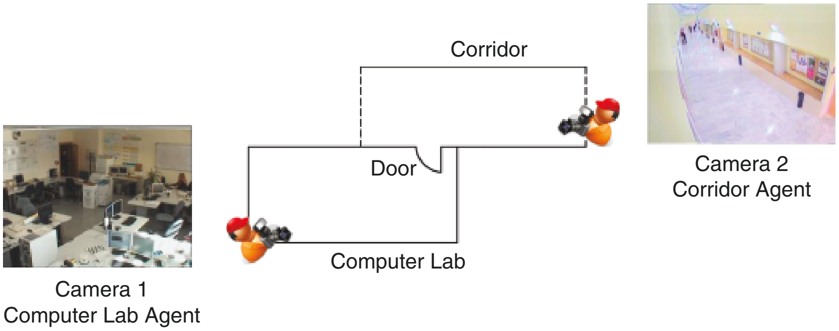


Figure 5: *Indoor scenario. Two camera agents: camera 1 is guarding a computer lab with two doors; camera 2 is placed outside the room, in the access corridor.*

authors' web page[5]. It is interesting to note that additional axioms or rules to calculate complex properties of tracks (distances, proximity, etc.), as well as spatial relationships (inclusion, adjacency, etc.), could be considered and created in TREN. Visual complex properties can be calculated from ontological tracking data by using appropriate formalisms, such as lambda calculus, as described in Gómez-Romero *et al.* (2009d). Regarding ontological representation and inference of spatial relationships, some recent works have proved that their direct management is impractical even for small knowledge bases (Stocker & Sirin, 2009), and that the expressiveness of OWL is not enough for encoding part of their semantics (Katz & Cuenca-Grau, 2005; Grütter *et al.*, 2008), especially if they are imprecise. These improvements remain as prospective directions for future work, as mentioned in section 6.

5. Use case

Let us suppose an indoor surveillance system to detect and track intruders inside the university facilities. The system encompasses two cameras covering a computer lab and an access corridor (see Figure 5). The most important challenge in this scenario is to guarantee tracking continuity for objects that leave the corridor to enter the computer lab. This is a particular case of intruder detection and tracking inside the whole building – the overall objective of the system. In this example, a person moves from the corridor to the computer lab. The corridor agent cooperatively informs the computer lab agent by sending all the available information about the associated track. The computer lab agent can merge this information with its own perceptions, which are incomplete due to occlusions and shadows inside the room, to improve its performance. This procedure is known as camera handover: a camera, which is tracking and object, delegates the responsibility of *handling* this object to another camera. Camera handover is supported by the Tren ontology, which is used to encode the track data contained in the messages exchanged by the two agents.

5.1. BDI representation

To model the BDI agents of the problem, we make the following assumptions:

1. There is a single intruder. The system would work with more than one intruder, but we simplify this condition to make the explanation easier.
2. The intruder moves from the corridor to the computer lab through the guarded door.
3. One camera observes the whole room and the other one the corridor.
4. Camera 2 is executing a tracking plan. Camera 1 is executing the surveillance plan, but not the tracking plan.

The system encompasses two agents: the corridor agent and the computer lab agent. Each

[5] *http://www.giaa.inf.uc3m.es/miembros/jgomez/ontologies/tren.owl*

agent manages two types of beliefs: tracking beliefs and contextual beliefs. Tracking beliefs are expressed with instances the TREN ontology. That is, the agents share the definition of the concepts and relations of the ontology, but each one has its own instantiation according to the current perceived tracks. Contextual beliefs express additional knowledge that may be useful in the handover. In this case, the corridor agent must know that the computer lab agent is guarding the door and vice versa. Additionally, it may be interesting to explicitly assert information about the static entities of the scene, for instance the location of the door itself. Contextual beliefs can be expressed as instances of the TREN ontology or, alternatively, with a more abstract ontology mapping real-world objects of the domain with physical properties captured by the cameras (size, position, colour, etc.) – see the next section.

Accordingly, the corridor agent manages the beliefs corresponding to: (i) the presence and the position of door1 (contextual); (ii) the proximity of the computer lab agent (contextual); and (ii) the presence and the position of track1 (tracking). Other beliefs should be incorporated into the knowledge base of the corridor agent to describe more contextual aspects of the scene (e.g. additional doors) and to manage other tracks. Figure 6 depicts the initial situation of the scenario. We include below an excerpt of the definition of the ontology instances describing the position of track1 in OWL 2 Manchester syntax (Horridge & Patel-Schneider, 2009):

```
Individual:  frame1
    Types:
        Frame

Individual:  unknown_frame
    Types:
        UnknownFrame

Individual:  track1
    Types:
        Track
    Facts:
        hasTrackSnapshot  track1_sn_1
```

The computer lab agent includes similar instances for describing doorA and the proximity of the corridor agent. Both agents must agree to the correspondence between door1 and doorA, which are the same object, and assert this equivalence in their knowledge base. Initially, the computer lab agent has no information about tracks, since it is not executing the tracking plan.

5.2. *Agent communication*

Communication begins when the corridor agent, which is executing the surveillance and the tracking plans, detects that an intruder is close to door1; i.e., that a fact stating that `(door1, ?t: close)` has been asserted in the belief base as a result of the tracking process (`?t` is a TrackSnapshot instance). Consequently, the corridor agent initiates a *warning about expected object* dialogue with the computer lab agent, since the intruder is likely to appear in its field of view. Notice that at this point, the computer lab agent, which is executing the surveillance plan, only has beliefs about contextual information.

The corridor agent sends a `cfp` message including the description of track1. This description, in principle, may include only information about the last properties of the track as calculated by the tracking algorithm – alternatively, a temporal window could be defined. In the simplest case, the corridor agent includes the property values associated with the last valid TrackSnapshot of the track. To some extent, the corridor agent *prunes* a section of the graph

```
Individual:  track1_sn_1
    Types:
        ActiveTrackSnapshot
    Facts:
        has Actual Properties  track1_sn_1_properties,
        is Valid In Begin  frame1,
        is Valid In End  unknown_frame

Individual:  track1_sn_1_properties
    Types:
        ActualProperties
    Facts:
        has Position  track1_sn1_position
        hasSize  track1_sn1_size
        has Color  track1_sn1_color
        hasVelocity  track1_sn1_velocity

Individual:  track1_sn1_position
    Types:
        Position
    Facts:
        positionValue  p1

Individual:  p1
    Types:
        geom:  2DPoint
    Facts:
        x  135,
        y  140
```

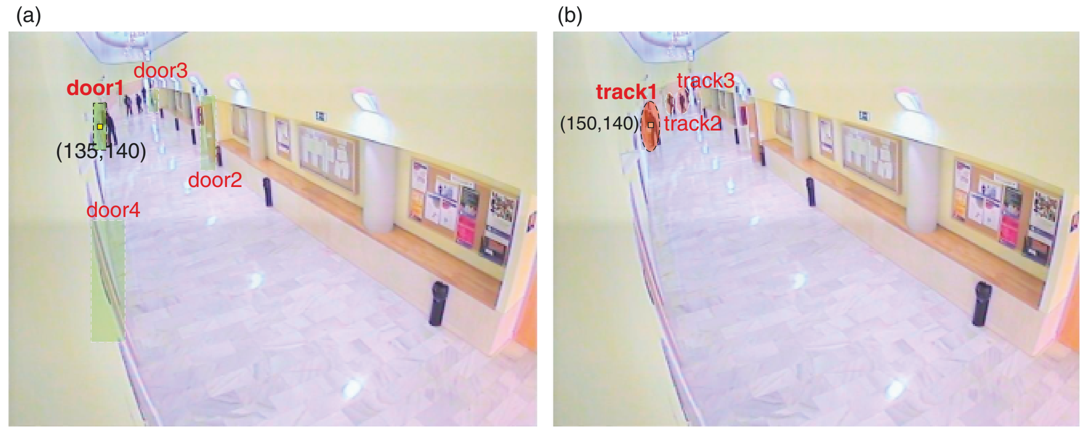


Figure 6: *Initial beliefs of the corridor agent (contextual and tracking).*

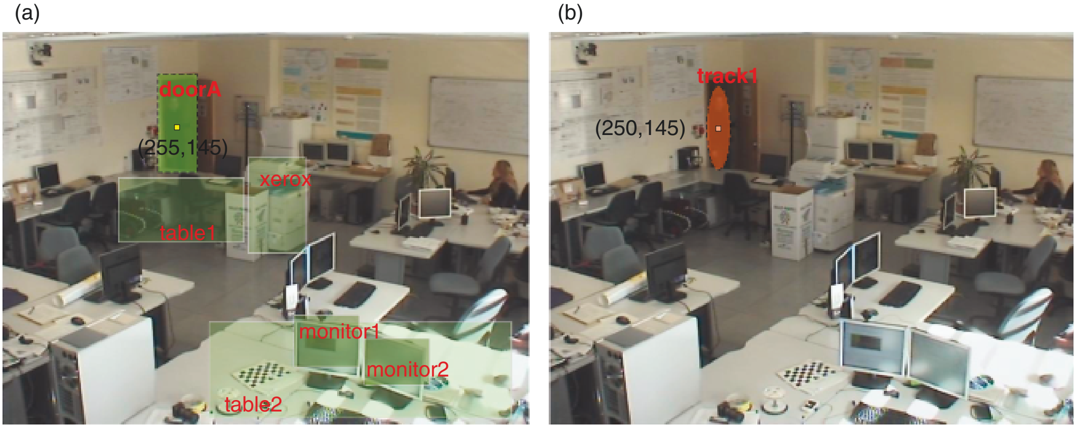


Figure 7: *Beliefs of the computer lab agent (contextual and tracking) after starting the tracking plan.*

composed of the ontology individuals associated with track1_sn_1 and sends it as the content of the `cfp`.

The computer lab agent notices the advice from the corridor agent: according to the property values asserted in the message sent by the corridor agent, the computer lab agent is now expecting an intruder to enter through doorA. This information is acknowledged back to the corridor agent with a propose message. This message may include some additional conditions that must be fulfilled to start the tracking. These conditions may refer to the track itself (e.g. size, colour) or to contextual conditions (e.g. the time of the day). The corridor agents agrees with the intentions of the computer lab agent and sends back an `accept-proposal` message. After this dialogue, the computer lab agent starts a tracking plan in the specified conditions.

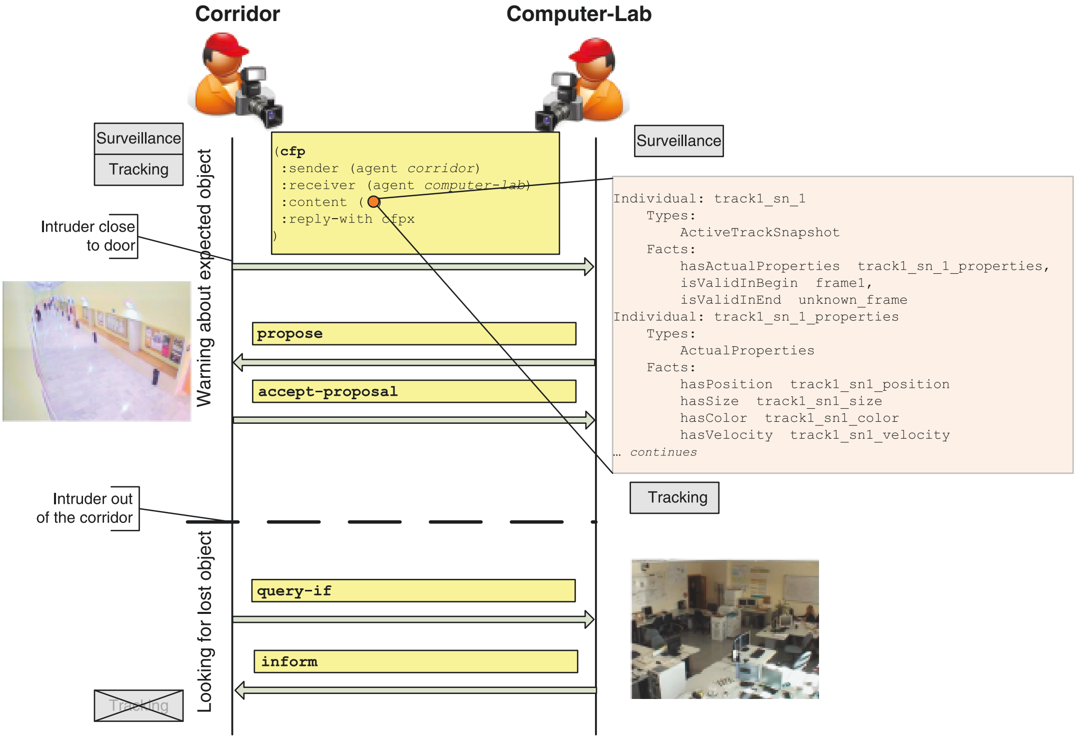
When the intruder comes into the room, the beliefs about tracked entities of the computer lab agent are updated to reflect the state of the scenario (Figure 7). The agent uses the information provided by the corridor agent in the previous dialogue to refine its estimation of the properties of the new track. This is especially interesting in this example, since errors due to the difficult conditions of the computer lab (furniture, illumination, etc.) are reduced with the use of redundant information.

At the same time, the corridor agent loses the track, because it is no longer inside its field of view. Therefore, it initiates a *looking for lost object* dialogue to ask neighbour agents about the intruder. The corridor agent sends a `query-if` message to the computer lab agent with a description of the lost track, which is actually quite similar to the one included in the previous `cfp` because track changes are minimum. The computer lab agent can easily identify that the corridor agent has lost the track corresponding to the intruder that it is now detecting. Hence, the computer lab agent sends back an `inform` message to notify the corridor agent that it is managing this track and to inform about track-updated properties; i.e., the instances asserted in his belief base related to the current snapshot of the missing track.

The complete sequence of events and messages is depicted in Figure 8.

5.3. *Supporting high-level information fusion*

As mentioned in the introduction, the ontological representation of tracking facts can be used not only to communicate information between camera agents, but also to initiate more complex high-level information fusion procedures. An extended development of a camera agent could implement an *a posteriori* schema for context information exploitation, as described in Gómez-Romero *et al.* (2010). This schema essentially proposes to implement a knowledge processing layer on top of the tracking procedure. In this layer, more abstract ontologies are used to describe more abstract entities; an ontology of an upper



```
(cfp
 :sender (agent corridor)
 :receiver (agent computer-lab)
 :content (
 :reply-with cfpx
)
```

```
Individual: track1_sn_1
    Types:
        ActiveTrackSnapshot
    Facts:
        hasActualProperties  track1_sn_1_properties,
        isValidInBegin  frame1,
        isValidInEnd  unknown_frame
Individual: track1_sn_1_properties
    Types:
        ActualProperties
    Facts:
        hasPosition  track1_sn1_position
        hasSize  track1_sn1_size
        hasColor  track1_sn1_color
        hasVelocity  track1_sn1_velocity
… continues
```

Figure 8: *Dialogues between the corridor agent and the computer lab agent to handle continuity of intruder tracking.*

abstraction level is *grounded* to an ontology of a lower abstraction level. For example, an ontology for scene objects can be defined. This ontology will define a property to associate instances of scene objects (e.g., *people*) to the actual track instances stored as agent's beliefs. Thus, information at this level is described in terms of objects, instead of in terms of tracks, but the association between them is purposely represented. Accordingly, a more abstract ontology could be defined to represent scene situations; these situations would be grounded to the involved objects, represented in the lower level scene objects ontology, which in turn is grounded to the track information ontology. Therefore, the TREN ontology is the lowest level ontology and allows for making a correspondence between cognitive and perceived entities.

We have developed a reference version of such ontological multi-level representation[6]. For a more extensive description of these ontologies, see (Gómez-Romero *et al.*, 2009b). We propose a set of ontologies structured according to the JDL model that include very general concepts and relations to represent knowledge in the surveillance domain. Particular applications are expected to define more precise concepts and relations within this framework. For example, it may be interesting to define a concept to represent a Column as a specialization of StaticObject and OcclusiveObject, which are concepts of the scene objects ontology. In the presented use case, a high-level ontology to represent the computer lab should include concepts and relations to describe the mentioned entities.

Standard ontology reasoning procedures can be performed within the ontologies to infer additional knowledge from the explicitly asserted facts. By using a DL reasoning engine, tasks such as classification or instance checking can be performed. These procedures can be considered as intra-ontology deductive reasoning,

[6] *http://www.giaa.inf.uc3m.es/miembros/jgomez/ontologies/*

since generally they obtain knowledge at the same abstraction level. For instance, extending the properties of OcclusiveObjects to their subclasses is an example of deductive reasoning. This is very useful to define very general rules that apply to more specific entities; e.g. 'do not delete tracks related to objects near an occlusive object'. In the use case, we could define the *near* property as transitive to avoid asserting exhaustively object location in the scenario.

In addition to this classical DL deductive reasoning, abductive reasoning procedures can be implemented. Scene recognition is a paradigmatic case of abductive reasoning, since it takes a set of facts as input (*observations*) and finds a suitable hypothesis that explains them (*interpretations*). In terms of the ontological infrastructure, this means to create new knowledge by applying abductive rules. These procedures can be considered as inter-ontology reasoning, since generally they obtain knowledge at a higher abstraction level. Abductive rules involve concepts defined in the ontologies; for instance, identifying the Hidding situation from the distance of an object to an occlusive object is an example of abductive reasoning. In the use case, an abductive rule would be defined to be triggered 'if a person, with an associated track information, is behind a column for more than a specified time'. Interestingly, abductive rules can be applied to accomplish both high-level information fusion and low-level tracking refinement; i.e., repectively, from perceptions to situation descriptions (bottom-up) and from situations to tracking corrections (top-down), as described in Gómez-Romero *et al.* (2009a). Currently, we are testing the implementation of such reasoning mechanisms, which are introduced in Gómez-Romero *et al.* (2009d).

6. Conclusions and future work

In this paper, we have proposed an ontology to represent tracking data in Computer Vision systems. The ontology is used in the CS-MAS framework to represent agents' beliefs and to communicate them by means of FIPA messages with contents described semantically. We have presented the structure of the ontology – which is publicly available – its use within the CS-MAS framework, and an example of communication of tracking data.

Endorsing data with semantics has several advantages. Communication is easily achieved among agents, since the contents of messages are expressed in the same well-defined language. Systems are more flexible, extensible and independent of the implementation technologies. The ontology also facilitates the development of extended functionalities for the vision system built on top of it, as well as the publication of tracking data.

We plan to continue this research work in various directions. First, we are applying the framework to problems more complex than the test case presented in this paper. This will allow us to obtain more precise measures of the accuracy and the performance of the system. The use of the ontology in different domains may imply further refinements or simplifications, in order to achieve a proper trade-off between the information shared by the agents and the computational resources needed to manage it – which are usually high in the case of large ontologies.

Additionally, we are studying how to represent more information, and specifically, uncertain and/or imprecise tracking data and spatio-temporal relations (e.g., *close*, *far*, *before*, *after*). It must be taken into account that standard ontologies do not provide support for this kind of knowledge; therefore, extended formalisms, such as fuzzy and probabilistic, would be needed. Closely related is the problem of conflict resolution, which occurs when an agent receives information that is not coherent with its own perceptions. Regarding knowledge representation, the uncertainty resulting from merging incoherent or contradictory pieces of knowledge should also be properly represented (in the agents' belief bases) and transmitted (through FIPA messages) by using the ontology.

Other interesting contribution would be the implementation of software tools for visualizing and exporting the data obtained by tracking algorithms and annotated with the ontology.

Such programs would be useful to review and test the accuracy of tracking processes with respect to a ground truth, both for expert and for non-expert users.

Moreover, as introduced in section 5.3, we are using the ontology as a basis for further high-level processing of visual data. We are testing an extension of the tracking system that uses context-knowledge to infer additional scene information at different fusion levels, including situation assessment and feedback. This context layer is built on top of the tracking data ontology, which is the first step in the evolution from numeric to symbolic information. An expected additional result in that regard is the exploitation of feedback procedures to enhance low-level tracking by reasoning from high-level scene interpretation.

Acknowledgements

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

Arndt, R., R. Troncy, S. Staab, L. Hardman and M. Vacura (2008) COMM: designing a well-founded multimedia ontology for the web, in *Proceedings of the 6th International Semantic Web Conference (ISWC 2007)*, Busan, South Korea, pp. 30–43.

Baader, F., D. Calvanese, D. McGuinness, D. Nardi and P.F. Patel-Schneider (2003) *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge: Cambridge University Press.

Baader, F., I. Horrocks and U. Sattler (2008) *Handbook of Knowledge Representation. Chapter Description Logics*, Amsterdam, The Netherlands: Elsevier, 135–180.

Bellifemine, F., G. Caire and D. Greenwood (2007) *Developing Multi-Agent Systems with JADE*, West Sussex, UK: Wiley Publishing.

Besada, J.A., J. Garca, J. Portillo, J.M. Molina and A. Varona (2005) Airport surface surveillance based on video images, *IEEE Transactions on Aerospace and Electronic Systems*, **41**, 1075–1082.

Braubach, L., A. Pokahr and W. Lamersdorf (2005) *Software Agent-Based Applications, Platforms and Development Kits. Chapter Jadex: A BDI-Agent System Combining Middleware and Reasoning*, Basel: Birkhäuser, 143–168.

Castanedo, F., J. García, M.A. Patricio and J.M. Molina (2010) Data fusion to improve trajectory tracking in a cooperative surveillance multi-agent architecture, *Information Fusion*, **11**, 243–255.

Erdur, R.C. and İ. Seylan (2008) The design of a Semantic Web compatible content language for agent communication, *Expert Systems*, **25**, 268–294.

François, A.R., R. Nevatia, J. Hobbs, R.C. Bolles and J.R. Smith (2005) VERL: an ontology framework for representing and annotating video events, *IEEE Multimedia*, **12**, 76–86.

Gangemi, A., N. Guarino, A. Oltramari and L. Schneider (2002) Sweetening ontologies with DOLCE, in *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, Sigüenza, Spain, pp. 223–233.

Gómez-Romero, J., J. García, M. Kandefer, J. Llinas, J.M. Molina, M.A. Patricio, M. Prentice and S.C. Shapiro (2010) Strategies and techniques for use and exploitation of contextual information in high-level fusion architectures, in *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, Edinburgh, UK.

Gómez-Romero, J., M.A. Patricio, J. García and J.M. Molina (2009a) Context-based reasoning using ontologies to adapt visual tracking in surveillance, in *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '09)*, Genoa, Italy, pp. 226–231.

Gómez-Romero, J., M.A. Patricio, J. García and J.M. Molina (2009b) Ontological representation of context knowledge for visual data fusion, in *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, Seattle, WA, USA, pp. 2136–2143.

Gómez-Romero, J., M.A. Patricio, J. García and J.M. Molina (2009c) Towards interoperability in tracking systems: An ontology-based approach, in *Methods and Models in Artificial and Natural Computation (A Homage to Professor Miras Scientific Legacy). Proceedings of the 3rd International Work-Conference on the Interplay between Natural and Artificial Computation (IWINAC 2009)*, Santiago de Compostela, Spain, pp. 496–505.

Gómez-Romero, J., M.A. Patricio, J. García and J.M. Molina (2009d) Towards the implementation of an ontology-based reasoning system for visual information fusion, in *Proceedings of the 3rd Annual Skövde Workshop on Information Fusion Topics (SWIFT 2009)*, Skövde, Sweden.

Grütter, R., T. Scharrenbach and B. Bauer-Messmer (2008) Improving an RCC-derived geospatial approximation by OWL axioms, in *The Semantic Web – ISWC 2008*, Karlsruhe, Germany: Springer Berlin/Heidelberg, 293–306.

HALL, D.L. and J. LLINAS (2009) Multisensor data fusion, in *Handbook of Multisensor Data Fusion*, M.E. Liggins, D. L. Hall and J. Llinas. Boca Raton, FL: CRC Press, 1–14.

HENDLER, J. (2001) Agents and the semantic web, *IEEE Intelligent Systems*, **16**, 30–37.

HITZLER, P., M. KRÖTZSCH, B. PARSIA, P.F. PATEL-SCHNEIDER and S. RUDOLPH (2008) OWL 2 Web Ontology Language primer, Online. W3C Recommendation. Available at http://www.w3.org/TR/owl2-primer/ (accessed 29 June 2011).

HOBBS, J. and F. PAN (2006) Time ontology in OWL, Online. W3C Working Draft. Available at http://www.w3.org/TR/owl-time/ (accessed 29 June 2011).

HORRIDGE, M., S. BECHHOFER and O. NOPPENS (2007) Igniting the OWL 1.1 touch paper: The OWL API, in *Proceedings of the OWL: Experiences and Directions Third International Workshop (OWLED 07)*, Innsbruck, Austria.

HORRIDGE, M. and P.F. PATEL-SCHNEIDER (2009) OWL 2 Web Ontology Language Manchester Syntax, Online. W3C Working Group Note. Available at http://www.w3.org/TR/owl2-manchester-syntax/ (accessed 29 June 2011).

HORROCKS, I. (2008) Ontologies and the semantic web, *Communications of the ACM*, **51**, 58–67.

HORROCKS, I. and P.F. PATEL-SCHNEIDER (2004) Reducing OWL entailment to description logic satisfiability, *Web Semantics: Science, Services and Agents on the World Wide Web*, **1**, 345–357.

KATZ, Y. and B. CUENCA-GRAU (2005) Representing qualitative spatial information in OWL-DL, in *Proceedings of the OWL: Experiences and Directions First International Workshop (OWLED 05)*, Galway, Ireland.

KOKAR, M., C.J. MATHEUS and K. BACLAWSKI (2009) Ontology-based situation awareness, *Information Fusion*, **10**, 83–98.

KOKAR, M. and J. WANG (2002). Using ontologies for recognition: an example, in *Proceedings of the 5th International Conference on Information Fusion (Fusion 2002)*, Vol. 2, Annapolis, MD, USA, pp. 1324–1330.

LEE, W., T. BÜRGER and F. SASAKI (2009). Use cases and requirements for ontology and API for media object 1.0, Online. W3C Working Draft. Available at http://www.w3.org/TR/media-annot-reqs/ (accessed 29 June 2011).

LLINAS, J., C. BOWMAN, G. ROGOVA, A. STEINBERG, E. WALTZ and F. WHITE (2004) Revisiting the JDL data fusion model II, in *Proceedings of the 7th International Conference on Information Fusion (Fusion 2004)*, Stockholm, Sweden, pp. 1218–1230.

MAILLOT, N., M. THONNAT and A. BOUCHER (2004) Towards ontology-based cognitive vision, *Machine Vision and Applications*, **16**, 33–40.

NEUMANN, B. and R. MÖLLER (2008) On scene interpretation with description logics, *Image and Vision Computing*, **26**, 82–101.

NOWAK, C. (2003) On ontologies for high-level information fusion, in *Proceedings of the 6th International Conference on Information Fusion (Fusion 2003)*, Vol. 1, Cairns, Australia, pp. 657–664.

NOY, N. and A. RECTOR (2006) Defining n-ary relations on the Semantic Web, Online. Available at http://www.w3.org/TR/swbp-n-aryRelations (accessed 29 June 2011).

PATRICIO, M.A., J. CARBÓ, O. PÉREZ, J. GARCÍA and J.M. MOLINA (2007) Multi-agent framework in visual sensor networks, *EURASIP Journal on Applied Signal Processing*, **2007**, 226–247.

PINZ, A., H. BISCHOF, W. KROPATSCH, G. SCHWEIGHOFER, Y. HAXHIMUSA, A. OPELT and A. ION (2008) Representations for cognitive vision, *ELCVIA. Electronic Letters on Computer Vision and Image Analysis*, **7**, 35–61.

RAO, A. and M. GEORGEFF (1995) BDI agents: from theory to practice, in *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, Cambridge, MA, USA, pp. 312–319.

REGAZZONI, C., V. RAMESH and G. FORESTI (2001) Special issue on video communications, processing, and understanding for third generation surveillance systems, *Proceedings of the IEEE*, **89**, 1355–1367.

SCHIEMANN, B. and U. SCHREIBER (2006) OWL-DL as a FIPA-ACL content language, in *Proceedings of the Workshop on Formal Ontology for Communicating Agents*, Malaga, Spain.

SNIDARO, L., M. BELLUZ and G.L. FORESTI (2007) Domain knowledge for surveillance applications, in *10th International Conference on Information Fusion (Fusion 2007)*, Quebec, Canada, pp. 1–6.

SNIDARO, L. and G.L. FORESTI (2007) Knowledge representation for ambient security, *Expert Systems*, **24**, 321–333.

STEINBERG, A.N. and C.L. BOWMAN (2004) Rethinking the JDL data fusion levels, in *Proceedings of the MSS National Symposium on Sensor and Data Fusion*, Columbia, SC, USA.

STEINBERG, A.N. and C.L. BOWMAN (2009) Revisions to the JDL data fusion model, in *Handbook of Multisensor Data Fusion*, M.E. Liggins, D. L. Hall and J. Llinas. Boca Raton, FL: CRC Press, 45–67.

STOCKER, M. and E. SIRIN (2009) PelletSpatial: A hybrid RCC-8 and RDF/OWL reasoning and query engine, in *Proceedings of the OWL: Experiences and Directions Sixth International Workshop (OWLED 09)*, Chantilly, Virginia, USA.

Valera, M. and S.A. Velastin (2005) Intelligent distributed surveillance systems: a review, *IEE Proceedings – Vision, Image, and Signal Processing*, **152**, 192.

Westermann, U. and R. Jain (2007) Toward a common event model for multimedia applications, *IEE Multimedia*, **14**, 19–29.

Yilmaz, A., O. Javed and M. Shah (2006) Object tracking: a survey, *ACM Computing Surveys*, **38**, 1–45.