17 Distributed Data and Information Fusion in Visual Sensor Networks

Federico Castanedo, Juan Gomez-Romero, Miguel A. Patricio, Jesus Garcia, and Jose M. Molina

CONTENTS

17.1 INTRODUCTION

Computer vision, and in particular multi-camera environments, has been widely researched over the recent years, thus leading to several proposals of multi-camera or visual sensor networks (VSNs) architectures (Valera and Velastin 2005). The aims of these systems are very different; to name some of them, there are examples in surveillance applications (Regazzoni et al. 2001), sport domains (Chen and De Vlesschouwer 2010), or ambient intelligence applications for elderly care (Zhang et al. 2010). Despite the specific goal of each system, all of them have to cope with a distributed architecture of visual sensors to acquire and process information from the environment. The obtained information must then be fused in order to generate a meaningful global picture of the environment. Since a distributed VSN can be applied to different domains/scenarios, a specific ontology provides meaning and sense of the information that the system uses for interpretation purposes.

This chapter explores the use of the multi-agent paradigm and ontology-based knowledge representation formalisms to perform distributed data and information fusion (DIF) in VSNs. The multi-agent paradigm, which has been widely applied in distributed systems, provides a theoretical and practical framework to allow communication and cooperation among the components of the system. For instance, in Lesser et al. (2003) several multi-agent protocols are presented to solve the task allocation problem in distributed sensor networks, but without visual capabilities.

Classical distributed visual systems work well for monitoring and surveillance tasks, but they can be improved using a multi-agent paradigm and ontology-based mechanisms. The underlying idea is to provide autonomous elements of the system with standard communication capabilities compliant to a content ontology in the process to achieve high-level data fusion.

The remainder of this chapter is organized as follows. The next section describes the main requirements and issues that should be taken into account when building VSNs. Section 17.3 introduces the application of multi-agent systems in visual sensor domains. Section 17.4 provides a description of a specific architecture to fuse data in a VSN. An example using this architecture is shown in Section 17.5. Finally, Section 17.6 presents some open research problems and prospective directions for future work.

17.2 VISUAL SENSOR NETWORKS

Modern VSNs involve the deployment of a number of cameras in a wide area and the management of these geographically distributed monitoring points. Third-generation video systems apply techniques that resemble the human intelligent process of surveillance, which activates certain cognitive abilities, to satisfy the challenges posed to modern security applications (Regazzoni et al. 2001). The most characteristic aspect of third-generation video systems is the use of physically distributed cameras able to locally run image-processing algorithms. Due to the huge amount of data, the natural processing architecture for a VSN is distributed (hierarchical or decentralized) with processors dedicated to each visual data stream in a first level, before the information is communicated through the network. The combination of multiple

viewpoints brings potential improvements to the reliability and accuracy of the results, although the existence of multiple cameras inevitably increases the complexity of the system. Although it is conceivable to achieve real-time performance with centralized processing, sending raw video streams to centralized servers is not practical, especially if the communication costs between nodes are accounted. Hence, local processing is necessary. Moreover, distribution increases system robustness and fault tolerance, since the same information may be captured and replicated at different points of the network.

17.2.1 Requirements and Issues

Two main requirements usually arise in distributed visual systems. First, it is necessary to implement suitable procedures to fuse local data (captured by single cameras) in order to obtain an integrated view of the situation while reducing bandwidth consumption. Second, coherence and scalability of the global system must be guaranteed with independence of the specific sensors and their configuration. This objective is difficult to accomplish when new heterogeneous cameras are incorporated to build a large and scattered network. Consequently, local data acquired by distributed video cameras must be combined to obtain a global understanding of the current scenario. Therefore, distributed systems for VSNs require techniques, algorithms, and procedures to solve the following issues.

17.2.1.1 Communication

Information acquired from each camera should be shared with others cameras and processing nodes, usually over a wired or wireless Internet Protocol (IP) network. The first decision in a multi-camera system is the physical installation of cameras. The number and placement of individual cameras have a great impact on system cost and capabilities. Since the main objectives are precise tracking of interesting objects, maximizing reliability and continuity of tracks, thus target-to-target or background-to-target occlusions must be minimized by using multiple cameras monitoring the same area from different viewpoints.

17.2.1.2 Camera Calibration

Information in the VSN must be expressed in a common reference frame. Camera calibration, or common referencing, is the process of transforming from the local coordinates of each camera to a global coordinate space. Calibration and synchronization can be done during an offline phase prior to system operation. This process is necessary to have a correspondence between the objects captured by different cameras. The resulting translation may include a reconstruction step to obtain a 3D representation of the 2D image. The most employed methods for camera calibration are those proposed by Tsai (1987), Heikkila (2000), and Zhang (2000). When the cameras have significant overlapping fields of views, the homograph between two corresponding image ground planes from two cameras can be computed by using target footprint trajectories and optimization techniques (Lee et al. 2000, Black and Ellis 2001). Typically, images of a calibration target (an object whose location and

geometry are known) are first acquired. Then, correspondences between 3D points on the target and their image pixels are obtained. This involves estimating the intrinsic and extrinsic parameters of the camera by minimizing the projection error of the 3D points on the calibration object. The Tsai camera calibration technique was popular in the past, but it requires a nonplanar calibration object with known 3D coordinates. Zhang (2000) proposed a more flexible planar calibration grid method in which either the planar grid or the camera can be freely moved. For multi-camera surveillance applications with little or no overlap areas between cameras, research has focused on automatically learning camera topology.

Some authors have proposed online calibration techniques. For instance, Javed et al. (2008) exploited the redundancy in paths that humans and cars tend to follow (e.g., roads, walkways, and corridors) by using motion trends and appearance of objects to establish correspondence. In Ellis et al. (2003) and Makris et al. (2004), authors used learned entry and exit zones to build the camera topology by exploiting temporal correlation of objects transiting between adjacent camera fields of view. In Pollefeys et al. (2009), a method is proposed to simultaneously compute the epipolar geometry and synchronization of cameras after considering the epipolar constraints that need to be satisfied by every camera pair.

17.2.1.3 Object Detection

Interesting objects must be identified in the sequence of images provided by the camera. There are various approaches to the detection of moving objects. *Temporal differencing* is based on calculating the pixel-by-pixel difference of various consecutive frames (Lipton et al. 1998). *Background subtraction* is based on subtracting the current snapshot pixel values with a predefined background image (Piccardi 2004). *Statistical methods* are a variation of basic background subtraction method. They are based on the difference of additional statistical measures (Wang et al. 2003). *Optical flow*, in turn, is based on the computation of the flow vectors of moving objects over time (Barron et al. 1994).

17.2.1.4 Object Tracking

Detected objects should be tracked over time by matching the detections between consecutive frames. Object tracking, which involves state estimation and data association, has been traditionally tackled by applying statistical prediction and inference methods. Some tracking methods in general DIF are distributed multiple hypothesis tracking (MHT) (Chong et al. 1990), distributed joint probabilistic data association (JPDA) (Chang et al. 1986), covariance intersection (CI)/covariance union (CU) (Julier and Uhlmann 2001), and distributed Kalman filter (Olfati-Saber 2007).

In the case of video data association, it is necessary that objects are robustly tracked in time, even though the image processing algorithms may fail to segment them as single foreground regions (blobs) in some intervals. Problems with object segmentation often occur (Genovesio and Olivo-Marin 2004) when (1) the object is occluded by another region, a fixed object in the scene, or other moving object; (2) the object image is split into fragments during image segmentation; (3) the images from different objects are merged because of their close or overlapping projections on the camera plane.

Classical data association techniques have been adopted and extended by computer vision researchers. The JPDA filter has been applied to 3D vision reconstruction (Chang and Aggarwal 1991, Kan and Krogmeier 1996). Cox (Cox and Hingorani 1996) proposed the first adaptation of Reid's MHT (Reid 1979) to visual data association problems. In more recent approaches (Khan et al. 2005, Cai et al. 2006, Liu et al. 2008), a Markov Chain Monte Carlo strategy is applied to explore the data association space in order to estimate the maximum a posteriori joint distribution of multiple targets. Other recent approaches (Fleuret et al. 2008) are based on a discretized occupancy maps in the real world onto which the objects are projected. As we shall explain in the following, the estimation process is very sensitive to particular conditions of the scenario. Thus statistical methods may be insufficient in VSNs, which requires the incorporation of additional information and knowledge in the process.

17.2.1.5 Classification

Object and activity recognition aim to determine the type of an object (e.g., car, human, aircraft) or the type of an activity (e.g., approaching, walking, manoeuvring). Depending on the specific application, classification can involve object type classification (car, human, aircraft, etc.) or activity classification based on the object movements. Recognition can be viewed as a probabilistic reasoning problem, in which case it is tackled through probabilistic models (Markov models, Bayesian networks, etc.) (Hongeng et al. 2004). It can also be modeled as a classification problem, in which case pattern recognition techniques (neural networks, self-organizing maps, etc.) (Hu et al. 2004) are employed.

17.2.1.6 Process Enhancement

Process enhancement, also known as active fusion, focuses on the implementation of suitable mechanisms that use the more comprehensive interpretation of the current situation obtained after fusing data to improve the performance of the previous tasks. Generally speaking, process enhancement improves a fusion procedure by using feedback generated at a more abstract level. For instance, the behavior of a tracking algorithm can be changed once a general interpretation of the scene has been inferred. When the system recognizes that an object is moving out of the camera range through a door, the tracking procedure will be informed to be ready to delete this track in the near future.

17.2.2 Related Research

A wide range of alternative architectures and algorithms for distributed camera systems have been proposed in the last decade. Cai and Aggarwal (1999) proposed a multi-camera framework for people tracking in outdoor environments. Mittal and Davis (2003) developed a multi-camera system for people tracking and action analysis.

Video surveillance and monitoring (VSAM), developed by Collins et al. (2001), is a system that addresses the problem of tracking multiple objects in a multi-camera scenario. VSAM presents the global picture of the environment to a human operator through a unified graphical user interface.

Snidaro et al. (2003, 2004) described a system for outdoor video surveillance in which data are acquired from different types of sensors (optical, IR, radar). In the first level, data are fused to perform the tracking of objects in each zone of the monitored environment. Next, this information is sent to higher levels to obtain the global trajectories of the objects. They employed an aspect ratio metric obtained for each detected object over all the sensors. The fused result is obtained by weighting each sensor's aspect ratio measurement. Analogously, Besada et al. (2005) proposed a distributed solution for airport surface traffic control based on a video network.

Matsuyama and Ukita (2002) developed a real-time multi-camera vision system in which the cameras are moved automatically with three degrees of freedom (pan, tilt, and zoom) according to the situation.

Typical examples of commercial surveillance systems are DETEC (DETEC Online) and Gotcha (GOTCHA Online). For outdoor applications, a representative example is the DETER system (Pavlidis et al. 2001). DETER reports unusual movement patterns of pedestrians and vehicles in outdoor environments such as car parks. In these conditions, the systems typically require a wide spatial distribution that implies camera management and data communication. Nwagboso (1998) proposes combining existing surveillance traffic systems based on networks of smart cameras. The term "smart camera" is normally used to refer to a camera that has processing capabilities (either in the same casing or nearby) and can autonomously perform event detection and event video storage.

In general, third-generation surveillance systems provide highly automated information, as well as alarms and emergencies management. This is the aim of CROMATICA (CROMATICA Online), a system for crowd monitoring and its successor, PRISMATICA (Velastin et al. 2005), a pro-active integrated system for security management. PRISMATICA, which is one of the most sophisticated surveillance systems of the recent years, is a wide area multi-sensory, multimodal distributed system. It receives inputs from closed-circuit television (CCTV), local wireless camera networks, smart cards, and audio sensors. Intelligent devices in the network process sensor inputs and send/receive messages to/from a central server module. Another important project is ADVISOR (Siebel and Maybank 2004), which aims to assist human operators by automatically selecting, recording, and annotating images containing events of interest. Although both systems are classified as distributed architectures, they have a significant difference: PRISMATICA employs a centralized approach which controls and supervises the whole system, whereas ADVISOR can be considered a semi-distributed architecture. In Yuan et al. (2003), an intelligent video-based visual surveillance system (IVSS) is presented. This system aims to enhance security by detecting certain types of intrusion in dynamic scenes. The system involves object detection and recognition (pedestrians and vehicles) and tracking, with an architecture similar to ADVISOR (Siebel and Maybank 2004).

Scalability has been specifically addressed by including new security devices or analysis modules after the initial deployment of the surveillance system. Within this context, service-oriented computing has been used to design a framework to deploy video surveillance applications (Enficiaud et al. 2006). The authors used this framework to detect and count people in monitoring environments.

One disadvantage of most classical systems is that they rely on expensive computational costs. This high processing load may be impossible to accomplish in real-time video applications, since image processing introduces a bottleneck due to the foreground/background subtraction algorithms. A second problem is that the employed algorithms usually rely on very strong statistical assumptions (such as Gaussian linear dynamic models of targets and noise), which unfortunately do not hold in several application domains. In video processing, statistical techniques have encountered practical limitations mainly due to the difficulty of obtaining analytical models of the source errors.

Researchers have proposed solutions to overcome the problems that usually arise when dealing with visual information. There is a growing interest in the design of open and flexible DIF software architectures and techniques that improve the classical approaches. One of the main challenges for achieving enough reliability in the information inferred from a visual network is the use of appropriate context representation and management formalisms in the fusion process. Also, the coherence in the network requires communication and coordination mechanisms to share information and carry out the necessary adjustments in the information derived.

Besides, distributed visual data fusion must address problems that are common to any distributed data fusion application. First of all, when dealing with images as an input source, it is very difficult to have a predefined model of sensor error and a priori detection probabilities (visual information may have problems with illumination changes, occlusions, etc.) Other problems with distributed solutions are the need of clock synchronization between sources, the presence of out of sequence measurement and data incest problems.

For these reasons, in this chapter we explore the use of multi-agent architectures in distributed fusion with specific reasoning procedures at the low-level (contextual) and high-level to obtain an appropriate interpretation of the environment. The use of ontologies is also considered to represent the exchanged information and formalize the exploitation of contextual information.

17.2.3 Context-Based Approaches to High-Level Information Fusion

Broadly speaking, high-level information fusion (HLIF) refers to those inferences developed by IF systems which correspond to a higher level of abstraction. Cognitive approaches to HLIF propose building a symbolic model of the world, expressed in a logic-based language, to abstractly represent the scene objects, events, and behaviors, as well as the relations among them (Vernon 2008). Such a model can be regarded as the mental representation of the scene gained by cognitive software agents. It may include both perceptions and more complex contextual information. Cognitive approaches are robust and extensible, but they require the development of suitable interpretation and reasoning procedures.

The use of symbolic models to acquire, represent, and exploit knowledge in IF, and particularly in visual IF, has increased in the last decade. Lambert (2003) highlights three requirements that are crucial to the implementation of model-based IF systems: (1) to discern what knowledge should be represented, (2) to determine which representation formalisms are appropriate, (3) to elucidate how acquired and

contextual inputs are transformed from numerical measures to symbolic descriptions, which is known as the grounding problem (Pinz et al. 2008).

Regarding selection of knowledge to be represented, there is a consensus about the importance of context knowledge in visual IF. Recently, researchers in IF have recognized the advantages of cognitive situation models, and have pointed out the importance of formal context knowledge to achieve scene understanding. Specifically, the last revision of the Joint Directors of Laboratories (JDL) specification highlights the importance of context knowledge (Steinberg and Bowman 2009), especially when visual inputs are to be interpreted (Steinberg and Rogova 2008). Henricksen (2003) defines context as *the set of circumstances surrounding* a task *that are potentially of relevance to its completion*. Kandefer and Shapiro (2008) extend this definition and state that context is *the structured set of variable, external constraints to some (natural or artificial) cognitive process that influences the behavior of that process in the agent(s) under consideration*.

To be consistent with this definition, we can consider that context in visual applications includes any external piece of knowledge used to complement the quantitative data about the scene computed by straightforward image-analysis algorithms. Context information (CI) is therefore an "external constraint" (because it is not directly acquired by the primary system sensors) that "influences the behavior" of the fusion process (since it is used to guide and support visual IF). Adapting the characterization by Bremond and Thonnat (1996), four sources of CI must be taken into account in visual DIF: (1) the scene environment: structures, static objects, illumination, and other behavioral characteristics, etc.; (2) the parameters of the sensor: camera, image, and location features; (3) historic information: past detected events; (4) soft information provided by humans.

Several representation formalisms have been proposed to be used in IF problems. Nevertheless, logic-based languages have received modest interest, in spite of their notable representation and reasoning advantages. Moreover, in this case most approximations have used ad hoc first-order logic representation formalisms (Brdiczka et al. 2006), which have certain drawbacks: they are hardly extensible and reusable, and reasoning with unrestricted first-order logic models is semi-decidable. Recently, there is a special interest in ontologies (Nowak 2003), since they overcome these problems. Current approaches are using ontologies to combine contextual and perceptual information, but there is still a lack of proposals that describe in detail how context knowledge can be characterized and integrated in general fusion applications.

At the low-level IF (i.e., JDL levels 0 and 1), one of the most important contributions is the Core Ontology for Multimedia (COMM). COMM is an ontology to encode MPEG-7 data at image level (i.e., JDL L0) (Arndt et al. 2007). It is represented with the Ontology Web Language (OWL), the standard proposed by the World Wide Web Consortium (W3C) (Hitzler et al. 2009). COMM does not represent high-level entities of the scene, such as people or events. Instead, it identifies the components of a 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).

Other proposals are targeted at modeling video content at object level (i.e., JDL L1). For example, a framework for video event representation and annotation is described in François et al. (2005). This framework includes two languages, namely the Video Event Representation Language (VERL) and the Video Event Markup Language (VEML). VERL defines the concepts to describe processes, such as entities, events, time, and composition operations; and VEML establishes an XML-based vocabulary to markup video sequences, such as scenes, samples, streams, etc. VEML 2.0 has been partially expressed in OWL. Other authors have discussed and improved this approach to support the representation of uncertain knowledge (Westermann and Jain 2007). Halfway between data and object level is the research work by Kokar and Wang (2002), who present a symbolic representation for the data managed by a tracking algorithm. In this approach, the data managed by a tracking algorithm are represented symbolically to solve the grounding problem and to support further reasoning procedures. The low-level ontologies presented in Section 17.4.2 are based in this notion. In addition, higher-level knowledge inferred by abductive reasoning is also considered in our proposal.

High-level IF issues (i.e., JDL L2 and L3) are being dealt with ontologies as well. Little and Rogova (2009) study the development of ontologies for situation recognition, and propose a methodology to create domain-specific ontologies for information fusion based on the upper-level ontology Basic Formal Ontology (BFO), and its sub-ontologies SNAP and SPAN, used for endurant (*snapshot*) entities and perdurant (*spanning*) processes, respectively. In Neumann and Möller (2008), the authors present an ad hoc proposal for scene interpretation based on Description Logics and supported by the reasoning features of the Renamed Abox and Concept Expression Reasoner (RACER) (Häarslev and Möller 2001). The authors also distinguish between lower-level representations and higher-level interpretations to avoid the grounding problem. The representation of high-level semantics of situations with a computable formalism is also faced in Kokar et al. (2009), where an ontology encoding Barwise's situation semantics is developed. The approach in Aguilar-Ponce et al. (2007) defines a multi-agent architecture for object and scene recognition in VSNs. In addition, the later authors propose the use of an ontology to communicate information between task-oriented agents, in a similar way as the proposal described in Section 17.4.1. A practical approach to surveillance is shown by Snidaro et al. (2007), who developed an OWL ontology enhanced with rules to represent and reason with objects and actors.

All these works focus on contextual scene recognition, but it is also interesting to apply this knowledge to refine image-processing algorithms (which corresponds to JDL L4), as described in Section 17.1. An approach to this topic is presented in Gómez-Romero et al. (2011). In this paper, the authors describe an ontology-based framework to support scene recognition and fusion process enhancement, and discuss contributions and drawbacks from an architectural and knowledge management point of view.

17.3 MULTI-AGENT SYSTEMS IN VISUAL SENSOR NETWORKS

Multi-agent systems have been proposed as a solution for distributed surveillance, since they naturally support coordination of multiple tasks aimed at the analysis of

object behaviors in dynamic and complex situations. Multi-agent systems are arguably well suited for the development of distributed systems in dynamic environments as VSNs. Agents have been applied in several approaches to identify faces and adapt the segmentation process in monitoring context, as discussed in Lee (2003).

Solving tracking tasks is one of the most studied problems by approaches that use agents to monitor objects. It is possible for agents to communicate and cooperate to monitor multiple objects simultaneously. A representative example of this approach was proposed by Remagnino et al. (2004), where they design the camera agent to calibrate the camera, track objects, and learning their behavior. The authors proposed a multi-agent architecture for visual monitoring where the agents are dynamically created when a new object is detected in order to cast the concept of agent to the detected objects. Similar proposals were later discussed in Garcia et al. (2005), which focuses on the communication messages exchanged between agents. The work in Castanedo et al. (2010) is also based on the application of multi-agent systems in a VSN. Recently, Albusac et al. (2010) also proposed a multi-agent architecture to incorporate expert domain knowledge into automatic monitoring and to provide a scalable and flexible solution tested in an urban traffic scenario.

As a matter of fact, the notion of agent suits very well to the concept of intelligent camera, since each software agent acquires and processes the visual images. On the one hand, nodes in the VSN are *autonomous*, in the sense that they have processing capabilities to acquire and process information in its field of view. On the other hand, the *social abilities* of agents provide the necessary means to share the visual information across the network and cooperate in the overall objective of the VSN. In order to avoid errors due to local knowledge of the world, nodes (developed as agents) establish social relations to build a global fused result depicting a more accurate and abstract view of the scenario.

In addition, agent-based *standard communication protocols* are the support to achieve interoperation with other systems at a high abstraction level. Last but not least, the existence of several multi-agent *frameworks*, which hide particular communication details, provides an easy way for developing distributed systems due to the loosely coupled architecture of multiple agents.

Ontologies can be used in such architecture to define the content language of agents' messages. The use of a common communication ontology facilitates agent interoperability, since the messages are expressed in the same well-defined language. This allows systems to be flexible, extensible, and independent of the implementation technologies. Moreover, sharing and reusing features of ontologies make them especially suitable for DIF in VSN. As mentioned before, VSN applications are highly context-dependent, but ontologies can be reused or extended to suit specific domain requirements. The agent communication ontology defines a set of concepts to describe the tracking information interchanged by the agents of the VSN. It behaves as an agreed vocabulary that allows tracking data to be represented in an abstract, common, and understandable way. Agents manage a local instantiation of the ontology, where individual ontologies corresponding to runtime scenario data are created. As we explain in the next section, ontologies are used in the architecture not only as a message content language but also to represent fused data and contextual knowledge.

17.3.1 Belief–Desire–Intention Paradigm

Multi-agent systems (Weiss 1999) can be divided into three different types: reactive, deliberative, and hybrid. The belief–desire–intention (BDI) paradigm is considered a hybrid architecture, since it divides the execution time of the system between deliberation and execution. The main difference with respect to the purely reactive architectures is that hybrid architectures spend more time reasoning to choose the next plan for execution. On the contrary, purely deliberative architectures follow a pure logic representation that requires an agent to manipulate symbols, and the percentage of time spent on the execution of the actions is less than the hybrid ones.

BDI paradigm has an explicit representation of the agent's notion following Bratman's theory of practical reasoning (Bratman 1987). The knowledge of an agent at any given time is based on the state of the BDI data structures. The belief data structure stores facts in a belief base acquired from the environment. Desire represents the final affairs that an agent wants to achieve. Finally, Intention describes specific plans that an agent has committed to execute in order to achieve its desires. Therefore, intentions should be consistent with the agent's desires. The BDI reasoning cycle must choose those plans for execution that match with the agent's desires, given the current belief. In this sense, the BDI architecture follows a similar reasoning process as the rule-based planning systems. However, multi-agent architectures also implement the social and communication capabilities required in any distributed system.

One of the advantages of using a multi-agent architecture is the separation between the management of the execution control and the reasoning mechanism, and plan execution is clearly separated inside the architecture. Therefore, there is no need to have an external management process.

17.3.2 Communication and Coordination

Agent communication in the VSN is the cornerstone to more complex DIF procedures. Communication mechanisms and protocols employed by the agents are usually based on the speech act theory (Searle 1970). To the speech act theory, spoken sentences in natural language are actions that produce changes in the receiver. Thus, in agent-based models, utterances are actions that result in changes in the internal state of the agents involved in the conversation. The messages sent by the agents are labeled using specific intention identifiers (e.g., *query* or *inform*). Exchanged information may range from essential data to complete acquired sequences, and from raw data to processed information. Besides, communication protocols can be based on pull messages (*ask* for information) or push messages (*provide* information).

The current standards for communication in multi-agent systems are defined in the Foundation of Intelligent Physical Agents (FIPA) specifications. Regarding message-passing, FIPA defines Agent Communication Language (ACL), a transport language that defines the format of the messages' envelope, a set of communicative acts, and a set of interaction protocols. ACL allows specifying the vocabulary to be used to encode agent contents. Traditionally, message semantics have been expressed in the FIPA Semantic Language (SL), a first-order logic derived language. The main

17.3.1 Belief–Desire–Intention Paradigm

Multi-agent systems (Weiss 1999) can be divided into three different types: reactive, deliberative, and hybrid. The belief–desire–intention (BDI) paradigm is considered a hybrid architecture, since it divides the execution time of the system between deliberation and execution. The main difference with respect to the purely reactive architectures is that hybrid architectures spend more time reasoning to choose the next plan for execution. On the contrary, purely deliberative architectures follow a pure logic representation that requires an agent to manipulate symbols, and the percentage of time spent on the execution of the actions is less than the hybrid ones.

BDI paradigm has an explicit representation of the agent's notion following Bratman's theory of practical reasoning (Bratman 1987). The knowledge of an agent at any given time is based on the state of the BDI data structures. The belief data structure stores facts in a belief base acquired from the environment. Desire represents the final affairs that an agent wants to achieve. Finally, Intention describes specific plans that an agent has committed to execute in order to achieve its desires. Therefore, intentions should be consistent with the agent's desires. The BDI reasoning cycle must choose those plans for execution that match with the agent's desires, given the current belief. In this sense, the BDI architecture follows a similar reasoning process as the rule-based planning systems. However, multi-agent architectures also implement the social and communication capabilities required in any distributed system.

One of the advantages of using a multi-agent architecture is the separation between the management of the execution control and the reasoning mechanism, and plan execution is clearly separated inside the architecture. Therefore, there is no need to have an external management process.

17.3.2 Communication and Coordination

Agent communication in the VSN is the cornerstone to more complex DIF procedures. Communication mechanisms and protocols employed by the agents are usually based on the speech act theory (Searle 1970). To the speech act theory, spoken sentences in natural language are actions that produce changes in the receiver. Thus, in agent-based models, utterances are actions that result in changes in the internal state of the agents involved in the conversation. The messages sent by the agents are labeled using specific intention identifiers (e.g., *query* or *inform*). Exchanged information may range from essential data to complete acquired sequences, and from raw data to processed information. Besides, communication protocols can be based on pull messages (*ask* for information) or push messages (*provide* information).

The current standards for communication in multi-agent systems are defined in the Foundation of Intelligent Physical Agents (FIPA) specifications. Regarding message-passing, FIPA defines Agent Communication Language (ACL), a transport language that defines the format of the messages' envelope, a set of communicative acts, and a set of interaction protocols. ACL allows specifying the vocabulary to be used to encode agent contents. Traditionally, message semantics have been expressed in the FIPA Semantic Language (SL), a first-order logic derived language. The main

drawback of SL is that it is undecidable in its general form; i.e., it is not guaranteed that all the inferences are computable in a finite time. Therefore, there is a growing interest in using formal ontologies as content languages (Hendler 2001, Schiemann and Schreiber 2006, Erdur and Seylan 2008), since they have appropriate computational properties and several supporting tools.

Ontologies can be accordingly defined to describe visual information exchanged by the agents of the VSN. In the simplest case, a suitable ontology can be created to represent tracking information. Such ontology would define a vocabulary including a set of concepts, relations, and axioms to describe tracking data. Agents manage a local instantiation of the ontology, where individual ontologies corresponding to the runtime data provided by the low-level tracking procedure are represented. Thus, the agents use the same vocabulary to interchange beliefs, which internally can be represented by using the ontology or not. Decoupling internal and external belief representations and the use of formal and standard languages facilitate the incorporation of heterogeneous elements to the VSN. In the most complex case, this ontology can include more abstract terms to represent objects, situations, or threats, and be the support of more sophisticated high-level fusion procedures, as described in the next section.

Besides communication, multi-agents also support the implementation of coordination schemes along communication protocols, in order to promote cooperation between agents and achieve better solutions. One of the most employed protocols for agent coordination is the contract-net (Smith 1980), which is mainly focused on task allocation problems. In a VSN, coordination mechanisms can be used to form smart camera coalitions, i.e., groups of sensors able to carry out complex processing tasks and collaborate with their neighbors. Another typical example of the application of agent cooperation in VSNs is camera handover (Patricio et al. 2007).

17.4 MULTI-AGENT APPROACH TO MANAGE DATA IN VSN

In the multi-agent approach for DDF in VSN, we can distinguish two main types of agents: sensor agents and fusion agents. Since the sources are completely distributed, but the fusion process is carried out by a centralized process level, a hierarchical and partially distributed architecture is proposed as is shown in Figure 17.1.

The figure shows two sensor agents and one fusion agent. However, it is possible to deploy several agents of each specific type. The only constraint is that a set of sensor agents are managed by a fusion agent following a hierarchical scheme. That is, the whole system has to include fewer fusion agents than sensor agents.

Sensor agents obtain the tracking information from the sensed environment through the acquired images and communicate the detected tracks to the fusion agent. So the external perception of each sensor agent is based on the processed images. The local perception of each sensor agent's environment is stored in the belief base as agent's beliefs. The obtained images are processed following the previous steps: object detection, data association, and state estimation. On the other hand, the fusion agent receives the track information from sensor agents and fuses it to obtain a global view of the scenario. The more comprehensive knowledge of the current situation obtained after fusing data can be used to provide sensor

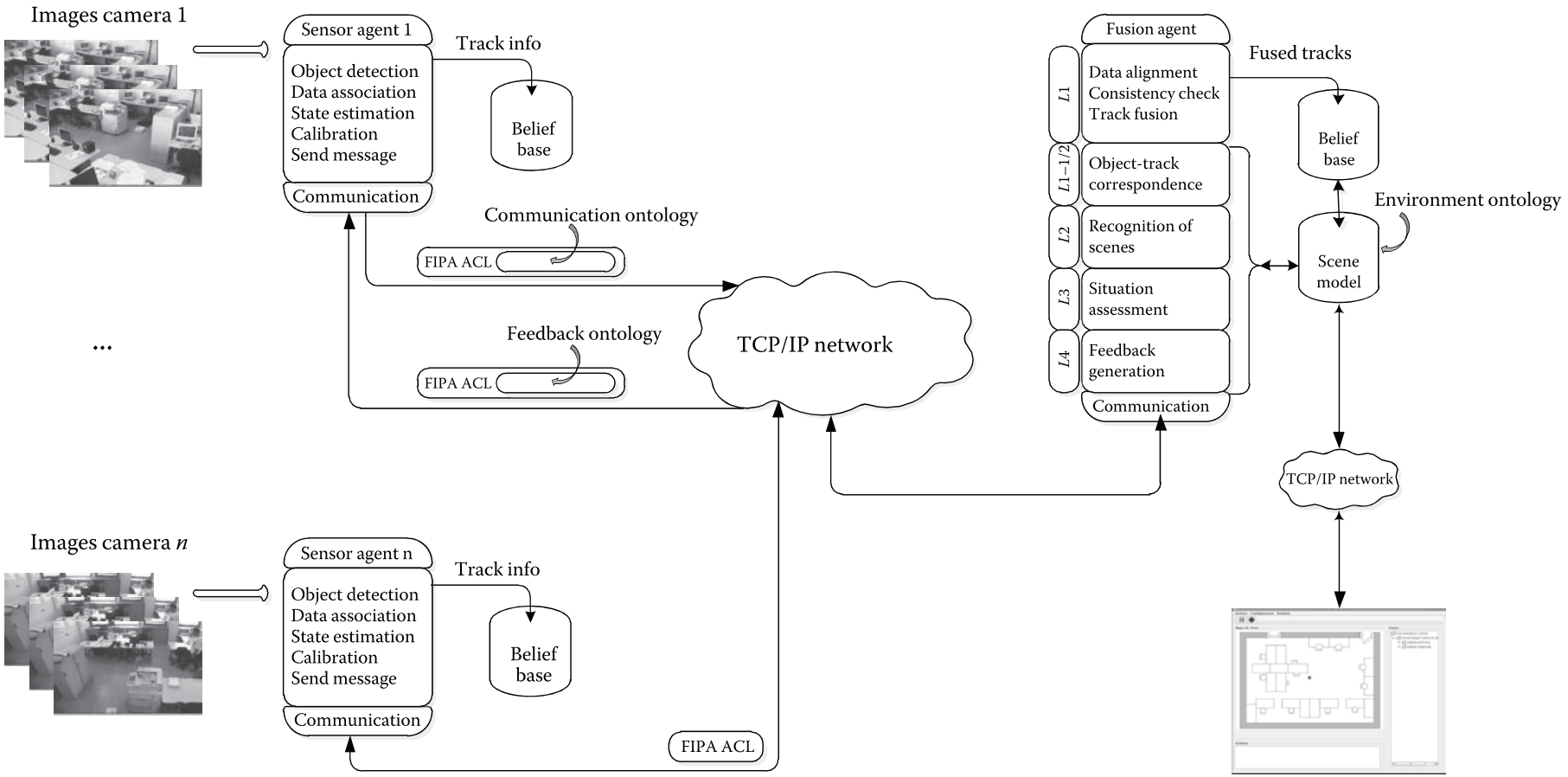


FIGURE 17.1 High-level hierarchical and partially distributed architecture.

agents with additional information, allowing them to correct their local knowledge. Communication between each sensor agent and the corresponding fusion agent is carried out by using the defined ontology as the content language in the FIPA ACL messages. Each agent (both sensor and fusion) is uniquely identified through its agent ID, which is composed of the IP address of the computer plus the agent platform and agent name. Next, the overall process is described in more detail.

17.4.1 Sensor Agents: Object Tracking

VSN data processing is performed by agents at two logical levels: (1) the tracking layer and (2) the BDI layer. First, each camera is associated with a tracking process. It sequentially executes various image-processing algorithms to detect and track all the targets within the local field of view. The tracking layer is arranged in a pipelined structure of several modules, as shown in Figure 17.2, which corresponds to the successive stages of the tracking process (Besada et al. 2005): (1) detection of moving objects, (2) blob-to-track multi-assignment, (3) track initialization/deletion, and (4) trajectory analysis.

The BDI layer uses an ontological model 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 numerical estimates from tracking. Therefore, the ontology is used for belief representation. This ontology, representing track information, can be also used for agent communication, as described in Section 17.3.2. Agent beliefs are represented as instances of the ontology, whereas desires and intentions are defined as plans following the JADEX format (Pokahr et al. 2005). We identify the following beliefs, desires, and intentions of camera-agents in a VSN:

Beliefs: Agent beliefs include information about the outside world, like objects that are being tracked (storing the location, size, trajectory, etc.), and geographic

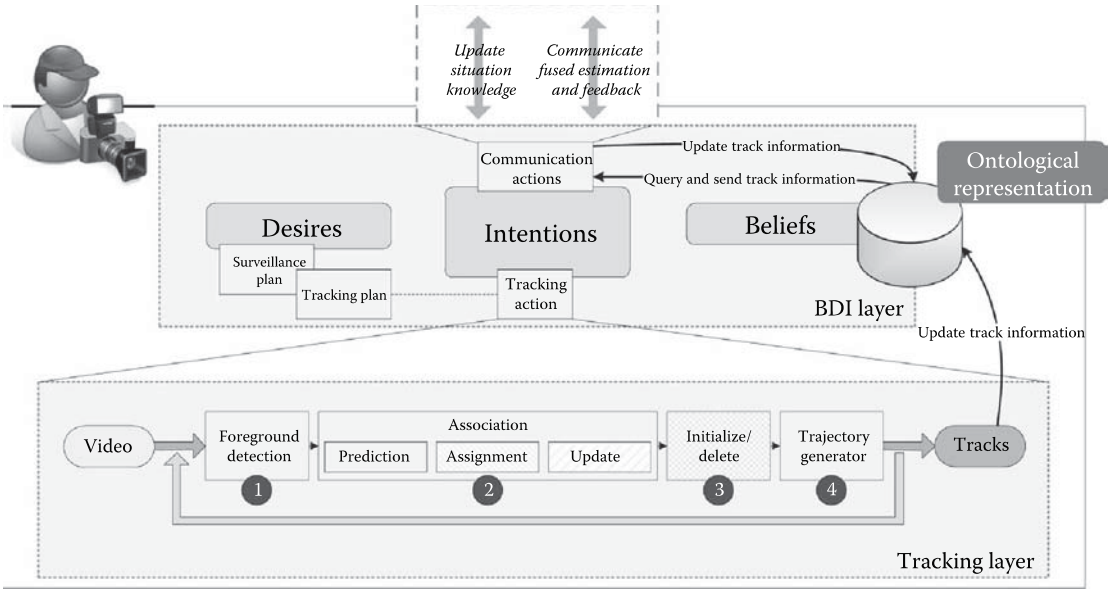


FIGURE 17.2 Sensor agent.

information about the camera itself, such as location, neighbor cameras, etc. The belief base of the agent is 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 will include convenient classes to describe tracks and track properties changing in time.

Desires: Since the final goal of agents is tracking the moving objects correctly, they have two main desires: permanent surveillance and temporary tracking. The surveillance plan is continuously executed. Sensor agents continuously capture images from the camera until an intruder is detected or announced by a warning from another agent. In this case, the tracking plan is triggered. The tracking plan runs inside a tracking process (implemented at the tracking layer), using the images from the camera until it is no longer possible. The tracking plan includes suitable actions to update beliefs of the agent, that is, to provide the track estimates to the BDI layer.

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. Agents send and receive messages carrying beliefs, which are represented with the ontology. Communication between sensor agents and fusion agents is performed by interchanging FIPA-compliant messages. The use of standard FIPA messages with content represented with the defined ontology promotes interoperability in the platform, as well as the incorporation of new heterogeneous agents. Two main types of interaction dialogs or conversations can happen between agents in the framework.

Update situation knowledge dialog: This interaction dialog sends to the fusion agent information about moving objects in the sensor agent field of view. The messages from the sensor agents include their local perceptions expressed as tracks and track properties represented in the communication ontology.

Communicate-fused estimation dialog: This interaction dialog sends to the sensor agent information and feedback about the global situation after data fusion is performed, according to the updates provided by the sensor agents.

17.4.2 Fusion Agents: Low- and High-Level Data Fusion, Context Exploitation, Feedback

The fusion agent processes the *update situation knowledge* messages which are sent by sensor agents and initiates the fusion process. The fusion agent first extracts suitable data from this formal representation and starts a low-level fusion process based on existing DIF algorithms. From this formal representation of the low-level fused tracks, a high-level fusion process is developed. High-level information fusion in the fusion agent has two objectives: (1) to obtain a high-level interpretation of the scene from the perceptions of the distributed sensors—i.e., to perform L1 to L3 fusion; and (2) to determine how the fusion processes might be changed to improve their performance—i.e., to perform L4 fusion.

Essentially, HLIF in the fusion agent is a model-building procedure, which results in the construction of an ontological instantiation that abstractly represents the fused

scene. We envision a knowledge model structured in five layers, from tracking data to impacts and threats:

Tracking data (L1). Output of the basic fusion algorithm represented symbolically. Examples include frames, tracks, and track properties (color, position, velocity, etc.)

Scene objects (L1 – L1/2). Objects resulting from making a correspondence between existing tracks and possible scene objects. For example, a track can be inferred to correspond to a person (possibly by applying CI). Scene objects include static elements which may be defined a priori and dynamic objects, which may be defined a posteriori. Examples include person, door, column, window, etc.

Activities (L2). Description of relations between objects that persist in time. Examples include grouping, approaching, picking/leaving an object, etc.

Impacts and threats (L3). Cost or threat value assigned to activities.

Feedback and process improvement (L4). Abstract representation of the suggestions provided to the tracking procedure.

An ontology of an upper abstraction level is based upon an ontology of a lower abstraction level. For example, the ontology for scene objects defines 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 tracks, but the association between them is purposely represented. In the same way, a more abstract ontology is defined to represent scene situations. These situations would be inferred from the relevant objects represented in the lower-level scene objects ontology, which in turn is related to the track information ontology. Therefore, the communication ontology is the lowest level ontology and allows for making a correspondence between cognitive and perceived entities.

The fusion process in the fusion agent is depicted in Figure 17.3. This figure represents the information processing flow: first from bottom to top, to interpret the scene; and second, from top to bottom, to generate feedback.

Scene interpretation is a paradigmatic case of abductive reasoning, in contrast to the Description Logics classical deductive reasoning. Abductive reasoning takes a set of facts as input and finds a suitable hypothesis that explains them (sometimes with an associated degree of confidence or probability). This is the case of scene interpretation: the objective is to figure out what is happening in the scene from the observations and the contextual facts. In terms of the fusion agent architecture, scene interpretation is an abductive transformation from instances of a lower-level ontology (representing perceived or contextual entities) to instances of a higher-level ontology. Abductive reasoning is not directly supported by ontologies (Elsenbroich et al. 2006), since monotonicity of ontology languages forbids adding new knowledge to the models while reasoning. Nevertheless, it can be simulated by using customized procedures or preferably by defining transformation rules in a suitable query language. The RACER inference engine, presented in Section 17.2.3, allows abductive reasoning, and therefore it may be a good choice to implement the reasoning procedures within the ontologies.

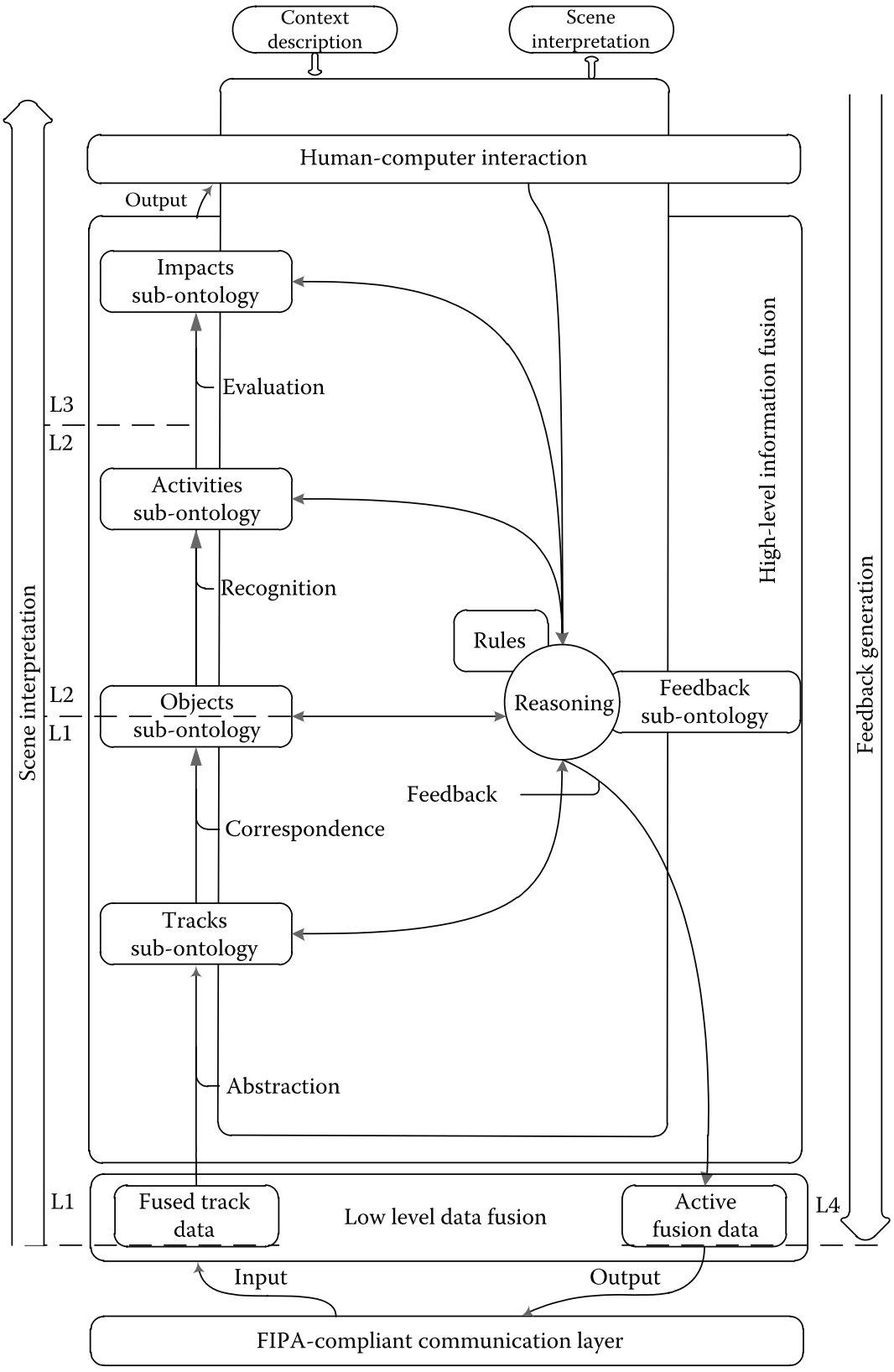


FIGURE 17.3 Fusion agent: low- and high-level fusion and feedback.

In the proposed architecture, abductive rules formally represent contextual, heuristic, and common sense knowledge to accomplish HLIF and low-level tracking refinement. Accordingly, we have two types of rules: bottom-up rules and top-down rules. On one hand, *bottom-up* rules are used in scene interpretation and obtaining instances of an upper-level ontology from instances of a lower-level ontology. For instance, some rules can be defined to identify objects from track measurements, i.e., to obtain instances of the scene objects ontology from instances of the tracking data ontology. An example rule may be: "create a person instance when an unidentified track larger than a predefined size is detected inside a region of the image." On the other hand, *top-down* rules create suggested action instances from the current interpretation of the scene, the historical data, and the predictions. These actions are used to adapt hypothesis at a lower-level to interpretations of a higher-level, which means the creation of instances of a lower-level ontology from instances of an upper-level ontology.

Eventually, top-down rules may create instances of the feedback ontology, which can be asynchronously returned to the sensor agent to update its knowledge. As a result of reasoning with the scene interpretation, active fusion information can be asynchronously returned to the sensor agent by starting a *communicate-fused estimation* dialog. These active fusion messages are also transmitted with the FIPA protocol and encoded with the communication ontology presented in Section 17.4.1.

17.5 APPLICATION EXAMPLE: INDOOR SURVEILLANCE

In this section, we will show how the framework presented in Section 17.4 is implemented in a specific application domain. Let us suppose an indoor surveillance system inside the university facilities aimed at tracking people and detecting interesting situations. We will focus on the computer laboratory, where three cameras are installed to cover the room area (see Figure 17.4). In this example, we have three sensor agents and one fusion agent. For the sake of simplicity, we will not consider additional cameras located at the nearby corridor. However, they can easily be incorporated to the framework and provide support for information handover when an individual enters the computer laboratory.

Before starting the processing, the framework must be configured. More precisely, the fusion agent must be informed of the positions of the cameras and provided with contextual information to be used in the fusion procedure. Once the framework has been configured, sensor agents start the execution of the *continuous surveillance* plan; i.e., agents process frames until the tracker detects a moving person in the room. Tracking data are encoded in the communication ontology and sent to the fusion agent by starting an *update situation knowledge* dialog. The fusion agent processes the tracking data obtained by the three cameras and combines them by applying a classical low-level fusion algorithm. This procedure results in updating the track data ontology, which triggers higher-level and contextual fusion procedures. Scene interpretation may lead to feedback generation to the sensor agents, which is returned back by starting a *communicate-fused estimation* dialog.

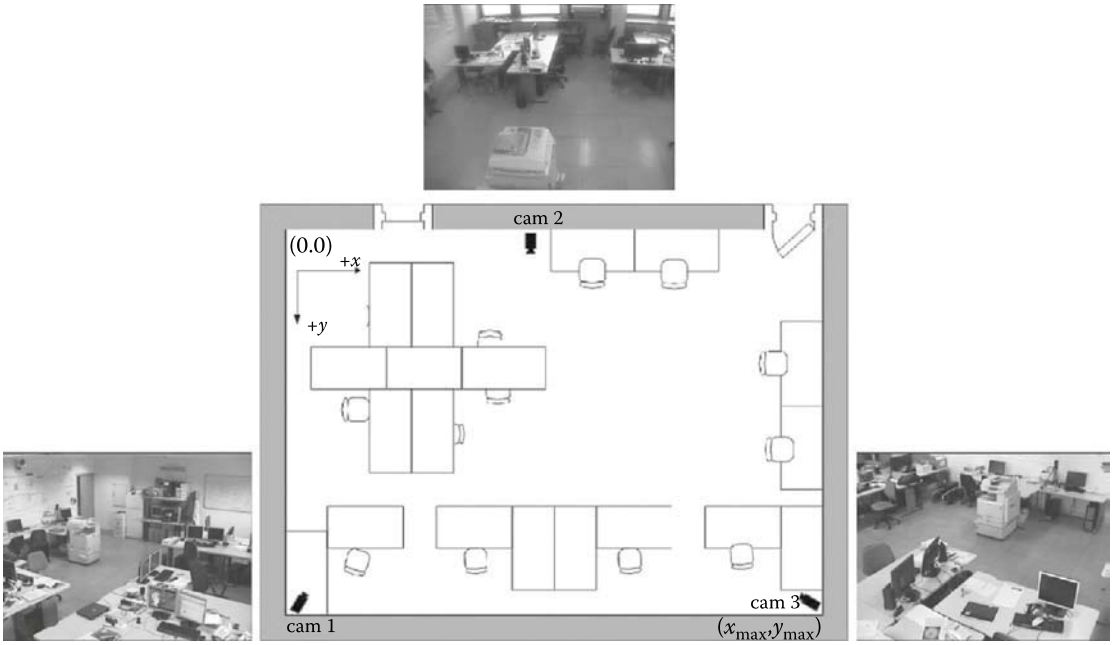


FIGURE 17.4 Computer laboratory scenario and cameras.

In the remainder of this section, we describe in more detail how these procedures are performed in the framework. This is not a comprehensive explanation of the implementation of such a system. Instead, we will make several assumptions to simplify the explanation of the system features in order to provide a general overview of the benefits of the approach and the open problems that remain to be solved in the future.

17.5.1 Framework Configuration: Camera Calibration and Context Definition

Camera calibration is achieved by applying the Tsai technique (1987). We manually mark some distinct points on the ground plane situated inside the overlapping area of the cameras. The homography matrix is calculated from the position of the distinct points in global and local coordinates. Linear optimization techniques are used to numerically calculate the values of the matrix. The homography matrix is used by the agents to transform from camera coordinates (as seen by sensor agents) to global coordinates (as seen by the fusion agent). Dynamic calibration techniques can be also applied, but for the sake of simplicity we will assume pre-calibration of the cameras (Figure 17.5).

After defining the common reference space, we use the ontological model to represent CI applicable to the scenario. Positions of the contextual entities are defined in global coordinates. To do this, we develop a specific ontology for surveillance based on the generic model presented in Section 17.4.2 to represent interesting entities of the surveillance domain, namely, the SURV ontology. This ontology defines the extensional knowledge of the application (i.e., concepts and relations). The intensional

FIGURE 17.5 An example of point correspondence in the three different views employed for the offline camera calibration phase.

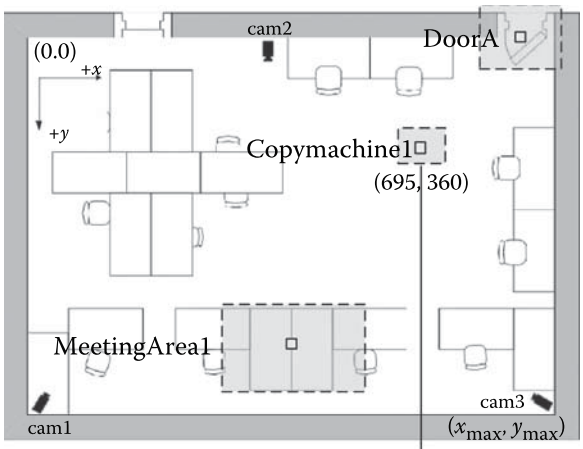
knowledge (i.e., instances) will be created as a result of the fusion procedure. The SURV ontology in this example imports the sub-ontologies of the generic model and specializes them, for instance, with additional

- Concepts:
  Objects: *Door, Person, Table, CopyMachine, MeetingArea*
  Scenes: *Approach, Meeting*
- Relations:
  inMeeting
- Axioms:
  $Person \sqsubseteq DynamicObject$ (a person is a dynamic object)
  $CopyMachine \sqsubseteq OccludingObject$ (a copy machine is an occluding object)
  $Table \sqsubseteq OccludingObject$ (a table is an occluding object)

The SURV ontology is used to annotate the scenario. Annotating the scenario means to create instances of the ontology describing static objects. Therefore, we initially insert instances in the ontology to indicate the position of the door, the tables, the copy machine, and the meeting area. Figure 17.6 depicts the correspondence between ontology instances and scenario information. We also show the OWL code corresponding to the definition of *copymachine1* as an instance of *CopyMachine* at position (695, 360) in global coordinates. Unfortunately, annotation must be performed manually. Further tools to support scenario annotation should be developed and learning procedures could be considered. These are interesting directions for future work.

After initialization, the SURV ontology is loaded into the reasoning engine (e.g., RACER). Contextual rules (abductive and deductive) must also be created in this step. Some simple example rules, expressed in plain text, are presented in the following. These rules are represented in a suitable rule language such as the previously mentioned nRQL.

- Object association:
  [Rule 1] If a track is bigger than (50×50) pixels, then it corresponds to a person
- Activity recognition:
  [Rule 2] If there are more than one person inside the meeting area for a while, a meeting is being held




```
<!-- copymachine1 instance -->
<owl:Thing rdf:about="#copymachine1">
   <rdf:type rdf:resource="#CopyMachine"/>
   <scob:hasObjectSnapshot rdf:resource="#osn_copymachine1"/>
</owl:Thing>

<!-- object snapshot of copymachine1 -->
<owl:Thing rdf:about="#osn_copymachine1">
   <rdf:type rdf:resource="&scob;SceneObjectSnapshot"/>
   <scob:hasObjectProperties rdf:resource="#copymachine1_props"/>
   <tren:isValidInEnd rdf:resource="&tren;unknown_frame"/>
</owl:Thing>

<!-- properties of copymachine1 snapshot (position) -->
<owl:Thing rdf:about="#copymachine1_props">
   <rdf:type rdf:resource="&scob;ObjectSnapshotProperties"/>
   <scob:OhasPosition rdf:resource="#copymachine1_position"/>
</owl:Thing>

<!-- copymachine1 position -->
<owl:Thing rdf:about="#copymachine1_position">
   <rdf:type rdf:resource="&scob;OPosition"/>
   <scob:OpositionValue rdf:resource="#p1"/>
</owl:Thing>

<!-- copymachine1 position value -->
<owl:Thing rdf:about="#p1">
   <rdf:type rdf:resource="&tren;2DPoint"/>
   <tren:y rdf:datatype="&xsd;float">695</tren:y>
   <tren:x rdf:datatype="&xsd;float">360</tren:x>
</owl:Thing>
```

FIGURE 17.6 Scenario annotation.

- Process enhancement and feedback:
  [Rule 3] If a person is close to an occluding object, sensor agents must be warned about a possible future occlusion
  [Rule 4] If a meeting is being held, do not care about the tracks associated to the people in the meeting to avoid confusion

17.5.2 Low-Level Information Fusion

Figure 17.7 depicts a scenario in which we have an individual moving around the following a predefined path (the ground truth is known a priori). The picture show the frames captured by the cameras at time $t=200$ s and the result of the background subtraction

FIGURE 17.7 Local tracking results obtained by sensor agents ($t = 200$ s).

procedure. The frames also include the bounding box calculated by each sensor agent as a result of the tracking procedure based on the local data in its field of view.

It can be seen that the results obtained by sensor agent 1 are not very accurate at this frame. Regarding sensor agent 1, while the x position of the center of the track is correctly calculated, the y position is moved up (in local coordinates). Regarding camera 3, both x and y positions of the track are misplaced, but this has no effect on the projection, since the individual's feet are correctly detected and positioned on the floor. The projection of the track position to the ground plane clearly shows this malfunctioning (Figure 17.8). The graphs depict the (x, y) positions in global coordinates estimated at each frame of the sequence with respect to the ground truth. Positions corresponding to the frames at $t = 200$ s are highlighted with a square.

Tracking information obtained by sensor agents is sent to the fusion agent, which performs a low-level fusion procedure to combine the tracks and correct sensor errors. We have used the algorithm presented in Castanedo et al. (2007). As explained, tracking information is encoded with the communication ontology and wrapped in FIPA-compliant messages. In this case, the results of the Fusion Agent outperform the local estimates, as depicted in Figure 17.9 where fused (x, y) positions on global coordinates at each frame are shown.

Fused tracking information is inserted into the HLIF knowledge model as instances of the tracking sub-ontology. This update may trigger further reasoning processes in the contextual layer, as described in Section 17.5.4. In addition, after detecting a deviation between local and fused estimates, the fusion agent may initiate an active fusion process and send appropriate feedback to sensor agents.

17.5.3 Contextual Enhancement to Tracking

In the previous example, estimation errors were the consequence of the limited information available. Thus, fusion significantly increased the accuracy of the system.

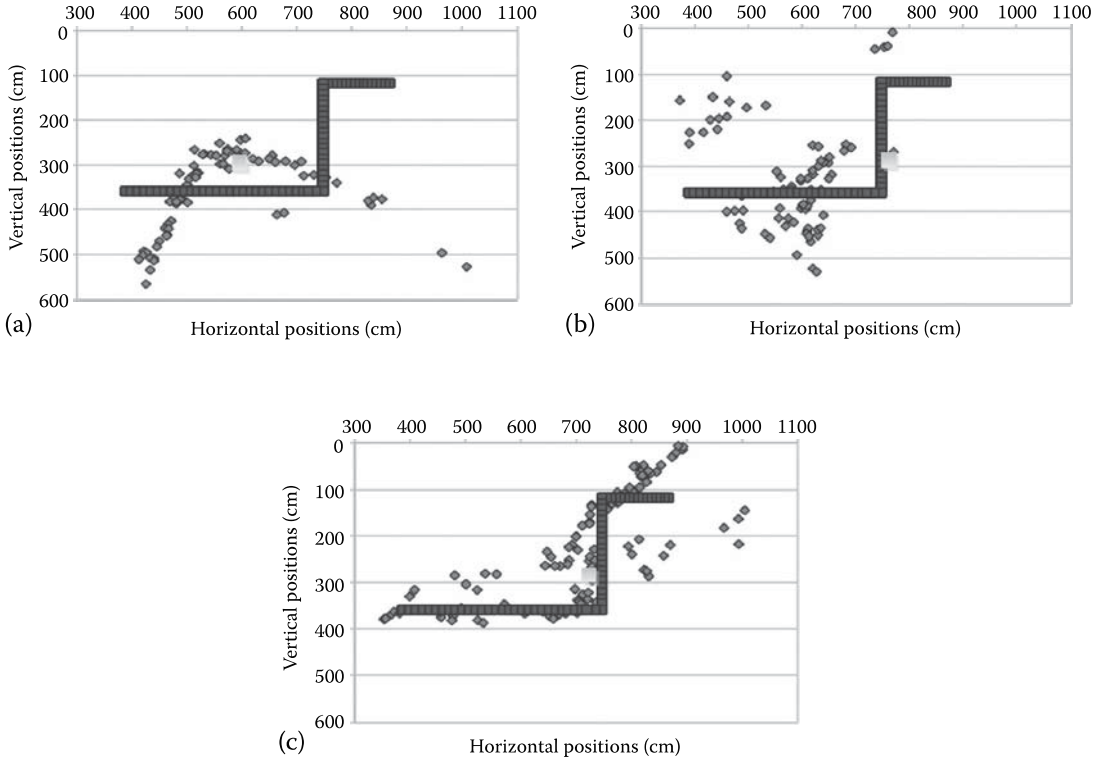


FIGURE 17.8 Local tracking results obtained by sensor agents compared to the ground-truth positions. (a) Sensor agent 1 (camera 1) (b) Sensor agent 2 (camera 2) (c) Sensor agent 3 (camera 3).

Nevertheless, in other cases classical data fusion procedures are insufficient to solve local tracking errors due to the inherent limitations of statistical tracking methods to adapt to complex situations.

For example, in Figure 17.10 we show the frames captured by the cameras at time $t = 180$ s and the (x, y) positions estimated at this frame in global coordinates.

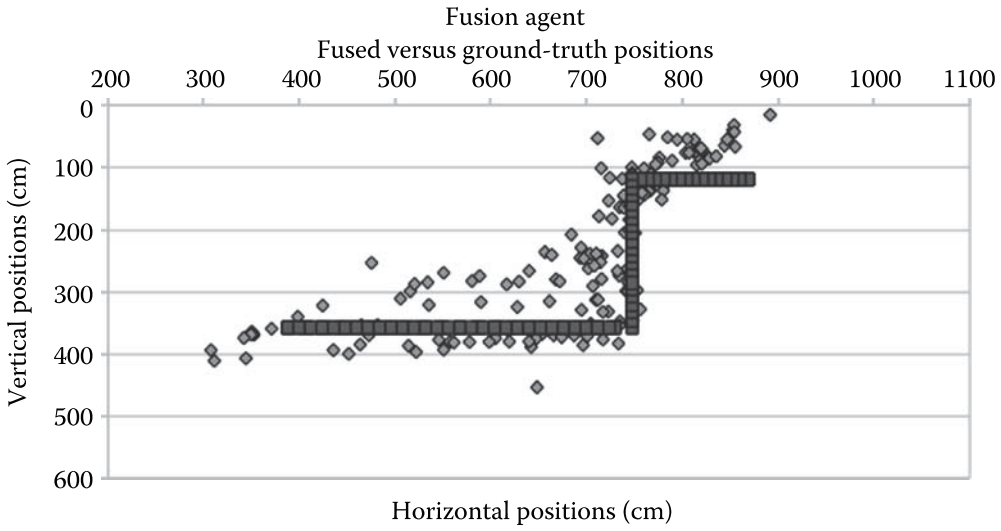


FIGURE 17.9 Fused tracking results obtained by fusion agent (from $t = 0$ to 200 s).

FIGURE 17.10 Local tracking results obtained by sensor agents ($t = 180$ s).

It can be seen that there is a significant error in the estimates of the three sensor agents. In this case, besides the previous difficulties (the individual is outside the field of view of cameras 2 and 3), there is an additional issue: a partial occlusion in camera 1. Partial occlusions result in track discontinuity, since hidden parts of the moving entities are not considered by the tracker and, therefore, track positions are misplaced.

Representation and reasoning with context knowledge in the fusion agent are applied to handle these situations. Scenario annotation is used to identify potential occlusive objects, contextual rules are fired when the conflictive situation is about to happen, and feedback is provided to the sensor agents to handle errors appropriately.

As a matter of example, let us suppose that the individual is being correctly detected by the tracker before $t = 180$ s. Fused information corresponding to this track would be consequently inserted into the HLIF knowledge model as instances of the track information sub-ontology. Rule 1 is triggered, and the track is identified as a person object by creating a proper instance in the object sub-ontology. In the next few frames, as the individual approaches the copy machine, the corresponding track information is updated, and eventually rule 3 is triggered. Consequently, an *expected occlusion* situation is created as an instance of the feedback sub-ontology. Subsequently, low-level fusion procedures and sensor agents may be notified about the situation by initiating a proper *communicate fusion estimate* dialog. If necessary, fused track information, encoded in the communication ontology, is sent back to the sensor agents by using FIPA-compliant messages. Low-level fusion procedures and sensor agents are responsible for handling the information properly. For instance, an appropriate action will be to incorporate track information to correct the Kalman filter matrix in order to avoid misplacing of the track position when the occlusion happens.

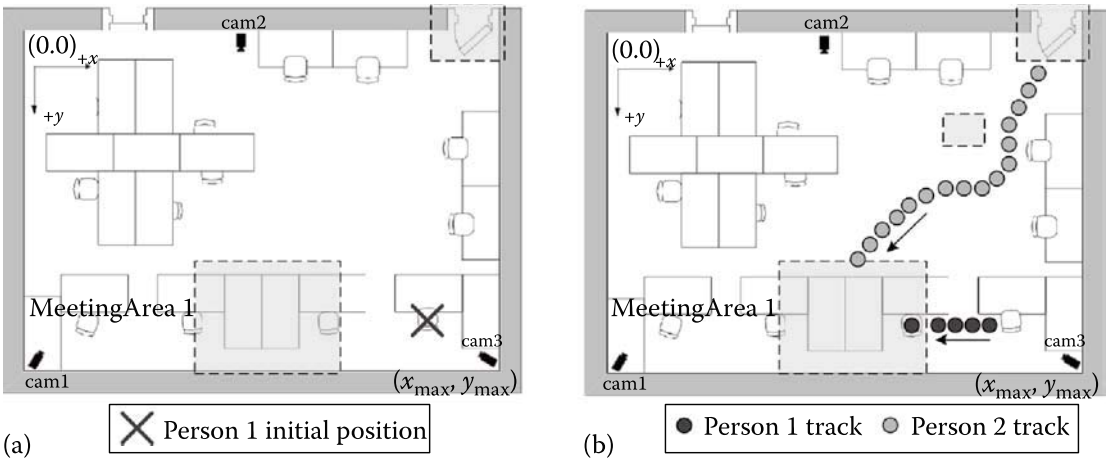


FIGURE 17.11 An example of detecting a meeting situation. (a) Person 1 is working without generating tracking updates and (b) activity in the meeting area results in a new detected situation.

17.5.4 Scene Interpretation

Let us suppose a situation in which we have an individual working on a desk of the computer laboratory (see Figure 17.11a). Tracking updates for these individuals are not sent to the fusion agent, because slight movements are not considered by the sensor agents. Next, one of the individuals (*person1*) stands up and moves into the meeting area. During this trajectory, sensor agents send information to the fusion agent, which updates the scene model. Some abductive and deductive reasoning procedures may be triggered as a result of ontology instance assertions, as explained before. Similarly, a second individual (*person2*) enters the room and moves to the meeting area. At this point, the current situation reflected in the ontological model is the one depicted in Figure 17.11b: we have two individuals labeled as persons who have entered the meeting area.

Consequently, rule 2 is triggered. A new *Meeting* instance is created in the activities sub-ontology, with *person1* and *person2* associated through the *inMeeting* property. This new *Meeting* instance fires rule 4. The aim of the rule is to prevent the agents from missing tracks corresponding to people who are close and probably overlapping. This feedback can be sent back to the sensor agents, which can handle this recommendation by stopping tracking in this area and storing track identifier and additional interesting track properties (e.g., predominant color), in order to identify tracks coming out of the meeting area.

17.6 SUMMARY AND FUTURE DIRECTIONS

More research works and implementations of general frameworks for visual DIF are needed to foster the creation of competitive solutions while cutting development costs in critical application areas. The first step toward domain-independent frameworks is to develop operational prototypes and to test them with existing data sets. The architecture proposed in Section 17.4 presents the overall picture of the

system, but real implementations will have to deal with several specific problems that are identified in the description. In Llinas (2010), the author envisions a possible approach to a general IF multi-layer framework with a front-end that manages hard and soft sensor inputs; an initial layer for detection, semantic labeling, and flow control, based on an intelligent repository of pluggable algorithms; a fusion layer, composed of several interrelated fusion nodes that process information at different JDL levels and incorporate CI to the process; and a presentation layer to convey the results through appropriate visualization interfaces. Such IF frameworks should provide an adaptable infrastructure where specific procedures can be easily reused and/or integrated, especially those based on artificial intelligence techniques, which are likely to play a key role in the next-generation fusion applications. We strongly believe that the multi-agent paradigm and ontologies as representation formalisms can be the theoretical support of such frameworks.

As for the specific design of the presented architecture, it is important to notice that we have proposed a hierarchical schema for DIF. We have limited data alignment at tracking level, but it should be possible to combine estimations performed by fusion agents at different levels in such a way that the system will be able to obtain a combined view of the scenario from the detected objects or the recognized situations, instead of only the track data. This will require further investigations both at data and process level, since it involves the formation of local coalitions of coordinated agents. Reputation mechanisms should also be taken into account to measure the confidence in the data provided by different sources, in order to achieve conflict resolution.

Another interesting research area is the incorporation of uncertain and vague information representation formalisms and reasoning procedures into the framework for visual HLIF. Classical ontologies do not provide support for this kind of knowledge, which is inherent to vision applications, and extensively, to IF applications. There are three main sources of uncertainty and imprecision in HLIF applications. Firstly, we have errors due to the imprecise nature of sensor data. They can be statistically modeled, but are affected by physical conditions. Secondly, there is uncertainty resulting from scene interpretation procedures; for example, when there is more than one object in the scene or the situation cannot be clearly discerned. Finally, there is uncertainty resulting from fusion procedures; for instance, data combination may be trusted to a certain degree. In addition, it may be interesting to add imprecise knowledge management features to the reasoning model in order to deal with vague spatiotemporal relations such as close, far, before, after, etc.

REFERENCES

Aguilar-Ponce, R., A. Kumar, J. L. Tecpanecatl-Xihuitl, and M. Bayoumi. 2007. A network of sensor-based framework for automated visual surveillance. *Journal of Network and Computer Applications*, 30(3):1244–1271.

Albusac, J., D. Vallejo, J. J. Castro-Schez, P. Remagnino, C. Gonzalez, and L. Jimenez. 2010. Monitoring complex environments using a knowledge-driven approach based on intelligent agents. *IEEE Intelligent Systems, Special Issue on Intelligent Monitoring of Complex Environments*, 25(3):24–31.

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

Barron, J., D. Fleet, and S. Beauchemin. 1994. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):42–77.

Besada, J., J. Garcia, J. Portillo, J. M. Molina, A. Varona, and G. Gonzalez. 2005. Airport surface surveillance based on video images. *IEEE Transactions on Aerospace and Electronic Systems*, 41(3):1075–1082.

Black, J. and T. Ellis. 2001. Multi camera image tracking. *Proceedings of the 2nd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*. Kauai, HI.

Bratman, M. E. 1987. *Intentions, Plans and Practical Reason*. Cambridge, MA: Harvard University Press.

Bremond, F. and M. Thonnat. 1996. A context representation for surveillance systems. *Proceedings of the Workshop on Conceptual Descriptions from Images at the 4th European Conference on Computer Vision (ECCV'96)*, Cambridge, U.K.

Brdiczka, O., P. C. Yuen, S. Zaidenberg, P. Reignier, and J. L. Crowley. 2006. Automatic acquisition of context models and its application to video surveillance. *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 1175–1178, Hong Kong, China.

Cai, Q. and J. K. Aggarwal. 1999. Tracking human motion in structured environments using a distributed camera system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1241–1247.

Cai, Y., N. de Freitas, and J. Little. 2006. Robust visual tracking for multiple targets. *European Conference on Computer Vision*, pp. 107–118, Graz, Austria.

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(3):243–255.

Castanedo, F., M. A. Patricio, J. García, and J. M. Molina. 2007. Robust data fusion in a visual sensor multiagent architecture. *The 10th International Conference on Information Fusion (FUSION 2007)*, Quebec, Canada.

Chang, Y. L. and J. K. Aggarwal. 1991. 3D structure reconstruction from an ego motion sequence using statistical estimation and detection theory. *IEEE Workshop on Visual Motion*, pp. 268–273, Princeton, NJ.

Chang, K., C. Y. Chong, and Y. Bar-Shalom. 1986. Joint probabilistic data association in distributed sensor networks. *IEEE Transactions on Automatic Control*, 31(10):889–897.

Chen, F. and C. De Vlesschouwer. 2010. Personalized production of basketball videos from multi-sensored data under limited display. *Computer Vision and Image Understanding*, 114(6):667–680.

Chong, C. Y., S. Mori, and K. C. Chang. 1990. Distributed multitarget multisensor tracking. In *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed., Vol. 1, pp. 247–295. Norwood, MA: Artech House.

Collins, R. T., A. J. Lipton, H. Fujiyoshi, and T. Kanade. 2001. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89:1456–1477.

Cox, J. and S. L. Hingorani. 1996. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18(2):138–150.

CROMATICA, Crowd Monitoring with Telematic Imaging and Communication Assistance. http://dilnxsrv.king.ac.uk/cromatica/ (accessed September 22, 2012).

DETEC, Detec Video Surveillance Software. http://www.detec.no/ (accessed September 22, 2012).

Ellis, T. J., D. Makris, and J. Black. 2003. Learning a multi-camera topology. *Proceedings of the Joint IEEE International Workshop VS-PETS*, Nice, France.

Elsenbroich, C., O. Kutz, and U. Sattler. 2006. A case for abductive reasoning over ontologies. *Proceedings of the OWL Workshop: Experiences and Directions (OWLED'06)*. Athens, GA.

Enficiaud, R., B. Lienard, and N. Allezard. 2006. Clovis—A generic framework for general purpose visual surveillance applications. *IEEE Workshop on Visual Surveillance*, pp. 177–184, Graz, Austria.

Erdur, R. C. and I. Seylan. 2008. The design of a semantic web compatible content language for agent communication. *Expert Systems*, 25(3):268–294.

Fleuret, F., J. Berclaz, R. Lengagne, and P. Fua. 2008. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282.

François, A., R. Nevatia, J. Hobbs, R. Bolles, and J. Smith. 2005. VERL: An ontology framework for representing and annotating video events. *IEEE Multimedia*, 12(4):76–86.

Garcia, J., J. Carbo, and J. M. Molina. 2005. Agent-based coordination of cameras. *International Journal of Computer Science and Applications*, 2(1):33–37.

Genovesio, A. and J. C. Olivo-Marin. 2004. Split and merge data association filter for dense multi-target tracking. *17th International Conference on Pattern Recognition*, Vol. 4, pp. 677–680. Cambridge, U.K.

Gómez-Romero, J., M. A. Patricio, J. García, and J. M. Molina. 2011. Ontology-based context representation and reasoning for object tracking and scene interpretation in video. *Expert Systems with Applications*, 38(6):7494–7510.

GOTCHA, Video Surveillance Software. http://www.gotchanow.com/ (accessed September 22, 2012).

Häarslev, V. and R. Möller. 2001. Description of the RACER system and its applications. *Proceedings of the International Workshop on Description Logics (DL2001)*, Stanford University, Stanford, CA.

Heikkila, J. 2000. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1066–1077.

Hendler, J. 2001. Agents and the semantic web. *IEEE Intelligent Systems*, 16(2):30–37.

Henricksen, K. 2003. A framework for context-aware pervasive computing applications. PhD thesis, University of Queensland, St. Lucia, Queensland, Australia.

Hitzler, P., M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph. 2009. OWL 2 web ontology language primer. In http://www.w3.org/TR/owl2-primer/ (Online, accessed on October 2011).

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

Hu, W., D. Xie, T. Tan, and S. Maybank. 2004. Learning activity patterns using fuzzy self-organizing neural network. *IEEE Transactions on Systems, Man and Cybernetics*, 34(3):1618–1626.

Javed, O., K. Shafique, and Z. Rasheed. 2008. Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2):146–162.

Julier, S. and J. Uhlmann. 2001. General decentralized data fusion with covariance intersection. In *Handbook of Multisensor Data Fusion*, 2nd edn., M. E. Liggins, D. Hall, and J. Llinas, eds., pp. 319–344. Boca Raton, FL: CRC Press.

Kan, W. and J. Krogmeier. 1996. A generalization of the pda target tracking algorithm using hypothesis clustering. *Signals, Systems and Computers*, 2:878–882.

Kandefer, M. and S. C. Shapiro. 2008. A categorization of contextual constraints. *Biologically Inspired Cognitive Architectures: Papers from the AAAI Fall Symposium*, pp. 88–93. Menlo Park, CA: AAAI Press.

Khan, Z., T. Balch, and F. Dellaert. 2005. Multitarget tracking with split and merged measurements. *Proceedings of the IEEE Conference on Vision and Pattern Recognition*, 1:605–661.

Kokar, M., C. Matheus, and K. Baclawski. 2009. Ontology-based situation awareness. *Information Fusion*, 10(1):83–98.

Kokar, M. and J. Wang. 2002. Using ontologies for recognition: An example. *Proceedings of the 5th International Conference on Information Fusion*, Vol. 2, pp. 1324–1330, Annapolis, MD.

Lambert, D. 2003. Grand challenges of information fusion. *Proceedings of the 6th International Conference of Information Fusion*, Vol. 1, pp. 213–220, Cairns, Australia.

Lee, R. S. T. 2003. iJADE surveillant—An intelligent multiresolution composite neuro-oscillatory agent-based surveillance system. *Pattern Recognition*, 36(6):1425–1444.

Lee, W., T. Bürger, and F. Sasaki. 2009. Use cases and requirements for ontology and API for media object 1.0. Retrieved from W3C Working Draft: http://www.w3.org/TR/media-annot-reqs/

Lee, L., R. Romano, and G. Stein. 2000. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):758–767.

Lesser, V., C. Ortiz, and M. Tambe. 2003. *Distributed Sensor Networks: A Multiagent Perspective*. Berlin, Germany: Springer.

Lipton, A., H. Fujiyoshi, and R. Patil. 1998. Moving target classification and tracking from real-time video. *Proceedings of the 4th IEEE Workshop Applications of Computer Vision (WACV 98)*, pp. 129–136, Princeton, NJ.

Little, E. G. and G. L. Rogova. 2009. Designing ontologies for higher level fusion. *Information Fusion*, 10(1):70–82.

Liu, J., X. Tong, W. Li, T. Wang, Y. Zhang, and H. Wang. 2008. Automatic player detection, labeling and tracking in broadcast soccer video. *Pattern Recognition Letters*, 30(2):103–113.

Llinas, J. 2010. A survey and analysis of frameworks and framework issues for information fusion applications. *Proceedings of the 5th International Conference on Hybrid Artificial Intelligence Systems conference (HAIS'10)*, LNCS 6076, San Sebastián, Spain.

Makris, D., T. J. Ellis, and J. Black. 2004. Bridging the gaps between cameras. *Proceedings of the Computer Vision and Pattern Recognition*, Washington, DC.

Matsuyama, T. and N. Ukita. 2002. Real-time multi-target tracking by a cooperative distributed vision system. *Proceedings of the IEEE*, 90(7):1136–1150.

Mittal, A. and L. Davis. 2003. M2 Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203.

Neumann, B. and R. Möller. 2008. On scene interpretation with Description Logics. *Image and Vision Computing*, 26(1):82–101.

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

Nwagboso, C. 1998. User focused surveillance systems integration for intelligent transport systems. In *Advanced Video-Based Surveillance Systems*, C. S. Regazzoni, G. Fabri, and G. Vernazza, Eds., Chapter 1.1, pp. 8–12. Boston, MA: Kluwer Academic.

Olfati-Saber, R. 2007. Distributed Kalman filtering for sensor networks. *Proceedings of the 46th Conference in Decision and Control*, pp. 5492–5498, New Orleans, LA.

Patricio, M. A., J. Carbó, O. Pérez, J. García, and J. M. Molina. 2007. Multi-agent framework in visual sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2001:1–21.

Pavlidis, I., V. Morellas, P. Tsiamyrtzis, and S. Harp. 2001. Urban surveillance systems: From the laboratory to the commercial world. *Proceedings of the IEEE*, 89(10):1478–1497.

Piccardi, M. 2004. Background subtraction techniques: A review. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4, pp. 3099–3104, The Hague, the Netherlands.

Pinz, A., H. Bischof, W. Kropatsch, et al. 2008. Representations for cognitive vision, ELCVIA. *Electronic Letters on Computer Vision and Image Analysis*, 7(2):35–61.

Pokahr, A., L. Braubach, and W. Lamersdorf. 2005. Jadex: A BDI reasoning engine. In *Multi-Agent Programming*, J. Dix, R. Bordini, M. Dastani, and A. Seghrouchni, eds. Dordrecht, the Netherlands: Kluwer.

Pollefeys, M., S. N. Sinha, L. Guan, and J. S. Franco. 2009. Multi-view calibration, synchronization, and dynamic scene reconstruction. In *Multi-Camera Networks: Principles and Applications*, H. K. Aghajan and A. Cavallaro, eds. Amsterdam, the Netherlands: Elsevier.

Regazzoni, C., V. Ramesh, and G. Foresti. 2001. Scanning the issue/technology: Special issue on video communications, processing and understanding for third generation surveillance systems. *Proceedings of the IEEE*, 89(10):1355–1366.

Reid, D. B. 1979. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854.

Remagnino, P., A. Shihab, and G. Jones. 2004. Distributed Intelligence for multi-camera visual surveillance. *Pattern Recognition*, 37(4):675–689.

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

Searle, J. R. 1970. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge, U.K.: Cambridge University Press.

Siebel, N. T. and S. J. Maybank. 2004. The advisor visual surveillance system. *Proceedings of European Conference on Computer Vision. Workshop Applications of Computer Vision*, Prague, Czech Republic.

Smith, R. G. 1980. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12):1104–1113.

Snidaro, L., M. Belluz, and G. L. Foresti. 2007. Domain knowledge for surveillance applications. *Proceedings of the 10th International Conference on Information Fusion (FUSION 2007)*, Quebec, Canada.

Snidaro, L, G. Foresti, R. Niu, and P. Varshney. 2004. Sensor fusion for video surveillance. *Proceedings of the 7th International Conference on Information Fusion (FUSION 2004)*, Stockholm, Sweden.

Snidaro, L., R. Niu, P. Varshney, and G. Foresti. 2003. Automatic camera selection and fusion for outdoor surveillance under changing weather conditions. *IEEE Conference on Advanced Video and Signal Based Surveillance*, Miami, FL.

Steinberg, A. N. and C. L. Bowman. 2009. Revisions to the JDL data fusion model. In *Handbook of Multisensor Data Fusion*, pp. 45–67. Boca Raton, FL: CRC Press.

Steinberg, A. N. and G. Rogova. 2008. Situation and context in data fusion and natural language understanding. *Proceedings of the 11th International Conference on Information Fusion (FUSION 2008)*, Cologne, Germany.

Tsai, R. 1987. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automaton*, 3(4):323–344.

Valera, M. and S. A. Velastin. 2005. Intelligent distributed surveillance systems: A review. *IEE Proceedings—Vision, Image, and Signal Processing*, 152(2):192.

Velastin, S. A., B. A. Boghossian, B. P. L. Lo, J. Sun, and M. A. Vicencio-Silva. 2005. Prismatica: Toward ambient intelligence in public transport environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35(1):164–182.

Vernon, D. 2008. Cognitive vision: The case for embodied perception. *Image and Vision Computing*, 26(1):127–140.

Wang, L., W. Hu, and T. Tan. 2003. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601.

Weiss, G. 1999. *Multi-Agent Systems. A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA: The MIT Press.

Westermann, U. and R. Jain. 2007. Toward a common event model for multimedia applications. *IEEE Multimedia*, 14(1):19–29.

Yuan, X., Z. Sun, Y. Varol, and G. Bebis. 2003. A distributed visual surveillance system. *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2003)*, Miami, FL.

Zhang, Z. 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.

Zhang, S., S. Mcclean, B. Scotney, X. Hong, C. Nugent, and M. Mulvenna. 2010. An intervention mechanism for assistive living in smart homes. *Journal of Ambient Intelligence and Smart Environments*, 2(3):233–252.