The Knowledge Engineering Review, Vol. 26:2, 191–208. © Cambridge University Press, 2011
doi:10.1017/S0269888911000026

On the use of agent technology in intelligent, multisensory and distributed surveillance

JOSÉ M. GASCUEÑA and ANTONIO FERNÁNDEZ-CABALLERO

Departamento de Sistemas Informáticos, Instituto de Investigación en Informática de Albacete, Universidad de Castilla-La Mancha, 02071-Albacete, Spain;
e-mail: Antonio.Fdez@uclm.es

Abstract

This article revises the state of the art of the application of agent technology within the scope of surveillance systems. Thus, the potential of the practical use of the concepts and technologies of the agent paradigm can be identified and evaluated in this domain. Current surveillance systems are noted for using several devices, heterogeneous in many instances, distributed along the observed scenario, while incorporating a certain degree of intelligence to alert the operator proactively to what is going on in the observed scenario and prevent the operator from having to observe the monitors continuously. The basic characteristics of the agents (autonomy, reactivity, proactiveness and social ability), along with multiagent systems' characteristics (distributed data management, low coupling, robustness, communication and coordination between autonomous entities), suggest that the agency is a good choice for solving problems which appear and are dealt with in surveillance systems.

1 Introduction

In the last few decades, the field of surveillance systems has captured the attention of the industry and investigation in the academic realm (López *et al.*, 2006a, 2007). This interest has recently grown to guarantee public safety due, in part, to the fatal attacks which took place in New York, Madrid and London. Hence, governments, enterprises and even citizens invest more and more money in equipment to make them feel safer. Several studies that analyze the state of the art of surveillance systems can be found in the literature (e.g. Hu *et al.*, 2004; Valera & Velastin, 2006; Kang & Deng, 2007; Kumar *et al.*, 2008; Haering *et al.*, 2008). These works focus on revising the architecture of surveillance systems and the algorithms used for visual surveillance. However, the contributions of agent technology are not mentioned.

There is no universally accepted definition for the term 'agent', but a wide range of perspectives exists in function of the application domain, the author and so on. Franklin and Graesser (1996) state: 'An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future'. Any agent, in accordance with this definition, satisfies the four properties as indicated next (Wooldridge & Jennings, 1995): (1) autonomy—agents operate without the direct intervention of humans or other agents and have some kind of control over their actions and internal state; (2) social ability—agents interact with other agents (and possibly humans) via some kind of agent-communication language (ACL); agents collaborate for the sake of performing tasks; (3) reactivity—agents perceive their environment (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet or perhaps all of these combined) and respond in a timely manner to changes therein; in order to respond effectively to changes, agents have to know at each instant their surrounding 'world'; (4) proactiveness—agents do not simply act in response to their environment; they are able to exhibit goal-directed behavior by taking the initiative.

The capability to communicate makes it possible for agents to work together to solve complex problems which cannot be dealt with by a single agent, this being the essence of multiagent systems (MAS; Huhns & Stephens, 1999). MAS are noted for the fact that they are made up of collections of potentially independent and autonomous agents, usually heterogeneous, which work together to solve a problem that goes beyond their individual capabilities. MAS are appropriate within domains in which the necessary knowledge to solve a problem is distributed along different places. The solution to the problem depends on the coordination of the tasks to be carried out by different entities with different capabilities, usually without the supervision of a single centralized coordinator. For example, defense applications are performed in highly decentralized and heterogeneous environments and/or require the incorporation of intelligent decision-making. These characteristics make the technologies, techniques and algorithms used within the scope of MAS adequate to be applied in military domain applications (Pechoucek *et al.*, 2008) such as logistics, manned and unmanned air traffic control, simulation and training. Likewise, on the one hand, Patricio *et al.* (2008) highlight the suitability of using an MAS for video surveillance because (1) the loose coupling nature of a multiagent architecture allows more flexibility in the communication process and (2) the ability to assign responsibilities to each agent is ideal to solve complex tasks in a surveillance system. These complex tasks entail the use of coordination and cooperation mechanisms and dynamic configuration, which are widely used in the MAS community (d'Inverno *et al.*, 1997). On the other hand, intelligence distribution in MAS allows one to deal with questions that turn up in the development of surveillance systems (bandwidth, productivity, speed, robustness, autonomy and scalability; Pavón *et al.*, 2007).

In summary, the basic characteristics of the agents (autonomy, reactivity, proactiveness and social ability), along with the characteristics of MAS (distributed data management, low coupling, robustness, communication and coordination between autonomous entities), suggest that they are good choices to solve the problems dealt with in surveillance systems. For this reason, the main goal of this article is to identify and evaluate the potential of the practical use of the concepts and technologies of the agent paradigm within the scope of surveillance systems. The rest of the article is organized as follows. First, the evolution of surveillance systems over time is shown. Next, the need to deal with the macro-task of video surveillance as a highly distributed and multisensory complex problem is emphasized. Afterward, a collection of works which use agent technology in related subjects to surveillance systems is summed up. Finally, some discussion and final conclusions are offered.

2 Surveillance systems

The functionality offered by surveillance systems improves as new technologies turn up and as users demand new solutions to their problems. From a historical point of view, it is acknowledged that the evolution of surveillance systems has gone through three generations (Valera & Velastin, 2006). In the first generation (1960–1980), closed-circuit television (CCTV) analog systems were used, which consisted of several cameras connected to a series of monitors. These systems do not process information and require a human operator to be permanently concentrated on analyzing the situations observed on the monitors. This practice is tedious, costly and inefficient, since the operator's visual attention to the video monitors decreases to unacceptable levels after a short period of time ($\sim 20\,\mathrm{min}$), thus increasing the likelihood of an anomalous situation not being detected. The analog signal also has the disadvantages of being very noisy, requiring larger amounts of bandwidth to be transmitted and more space to be stored than a digital signal.

However, in the second generation (1990–2000), the advances attained in digital video communication (e.g. digital compression, bandwidth reduction and robust transmission) are used to increase the efficiency of surveillance systems. CCTV systems are used together with computer vision technology to automatically process images, in order to detect alarm events proactively while recording. These semiautomatic systems require a robust tracking and detection algorithm to analyze their behavior. Although these systems represent a clear improvement regarding first generation systems, since they reduce the dependency on human operators to detect anomalous situations, the algorithms and techniques used are still not mature enough and can trigger a high number of false alarms. The data processing is carried out by a centralized system.

Finally, in the third generation (2000-today), a series of heterogeneous sensors (e.g. fixed cameras, pan-tilt-zoom (PTZ) cameras, audio sensors and RFID tags (radio-frequency identification) geographically distributed along the observed scenario are used. From the image processing point of view, these systems are based on distributing the processing capabilities and using embedded signal processing devices to gain from the advantages of the scalability and robustness of the distributed systems. The main problems that need to be solved in these systems are integrating data obtained from different sensors, establishing a correspondence of the signals in time and space and coordinating and distributing the processing task and video communication.

Currently, surveillance systems are fundamentally used for three reasons. First, to guarantee public safety in public spaces (e.g. train stations, subways or airports) and/or private spaces, detecting and preventing possible vandal or criminal attacks; second, in order to carry out a forensic analysis of the incidents that occurred; and third, to obtain statistics about people, vehicles and their behavior. The measurements gathered can be used, among other things, to plan the frequency of the city bus runs, to know the most popular time people go window shopping in big department stores, to know queue lengths and what the traffic flow is like on the roads, etc.

The construction of intelligent systems is reflected in many articles. In surveillance systems, the term 'intelligent' is used to express the systems' capability to identify automatically the target/object of interest and/or recognize its behavior (normal or suspicious). Among their capabilities, you may find the ability to detect movement, running, walking, loitering, fights, falls, people hiding, gesture recognition, leaving packages unattended, thefts, counterflow pedestrian traffic, entering forbidden areas of interest, recognition of different sports plays, recognition of suspicious behavior of people interacting with cars in a parking lot, vehicles performing wrong driving maneuvers, licence plate recognition or face recognition, following someone, being with someone, vandal scenes such as drawing graffiti, etc. If a surveillance system has all or a subset of these capabilities, then it can be labeled an intelligent surveillance system. The system is 'intelligent' since it carries out tasks on its own that a human being could carry out and it calls the attention of the security personnel only when necessary.

3 Multisensory and distributed surveillance

Third generation visual surveillance systems, as described earlier, can be described as multisensory and distributed systems. They are made up of a series of sensors distributed along the scenario. It is usual to find surveillance systems made up of devices which make it possible to recover a different flow of information (video, audio, images and scalar data) from the scenario to carry out an analysis and interpretation of the scene in real time (Cucchiara, 2005). For instance, in Trivedi *et al.* (2000), an architecture applied to highways and intersections is proposed. It is made up of several sensor clusters which can communicate with each other. A cluster includes microphones, omnidirectional CCD (charge-coupled device) cameras, infrared cameras and real-time range sensing cameras. The goal is to develop a detection, monitoring and recovery system for traffic incidents. Fusion of sensor data will enable the recognition of individual and group behavior such as a car going over the speed limit or several cars being involved in a high speed chase. PRISMATICA (Velastin *et al.*, 2004) is a large-scale multisensory distributed system which receives information from CCTV, local camera network, smart cards and audio sensors to improve surveillance in public transportation. For example, this system detects intrusions into forbidden places, counts passengers, measures the occupancy ratio in strategic areas, etc. There is a central computer in its architecture which controls and supervises the whole system. Hence, it is considered to follow a centralized approximation.

In Petrushin *et al.* (2006), a surveillance system is described which uses a network of different types of sensors to locate and track people in an indoor environment. To follow through with this reasoning, a Bayesian probabilistic framework is used which takes the information obtained by the network of sensors and incorporates it with the domain and contextual knowledge (e.g. location of doors, armchairs in a hall, hallways and cubicles), which are provided by the physical restrictions imposed by the local environment where the sensors are located and by the people involved in the surveillance task. In the system, there are components responsible for receiving signals from different webcams and storing the images in which movement was detected in the field of vision. Three other components are in charge of recognizing and storing the events detected by the infrared system, the results of a fingerprint reader and face recognition, respectively. Some or all of the characteristics extracted by the previous components are used, based on the aim of the application (count the number of people on the floor, recognize behavioral patterns in people, know people's location, and so on). Single Location Surveillance Point (SLSP) is a multisensory distributed surveillance software system for indoor environments (Räty *et al.*, 2008). Each sensor (fingerprint reader, camera, audio, network activity monitor) gets data from its environment and transmits them to the session server. The session server, which handles all connections between SLSP components, transmits the data received to the LDMS (Logical Decision Making Server). The LDMS automatically deduces the location of the surveillance point based on the data received. The LDMS gives the video recorder instructions based on the logical deductions obtained. Finally, the deductions are transmitted to the human surveillance operator of the SMSU (Security Manager Server) and to the end device of a nomad guard. SLPS has a centralized architecture. Moreover, it has the disadvantage of only being able to monitor a single location, for example, the entrance to a building.

However, in multisensory surveillance, there are also tasks that are focused exclusively on the use of several types of camera sensors. SensEye is a multitier network architecture designed for surveillance applications in wireless visual sensor networks (Kulkarni *et al.*, 2005). The sensors (cameras) are distributed among several layers, where each layer is made up of a homogeneous set of sensors. The architecture has three layers. Object detection takes place in layer 1. Object recognition takes place in layer 2. Tracking moving objects takes place in layer 3, as well as establishing communication with a base station. As we move upward a level, the number of sensors decreases and the requirements for power, capabilities and costs increase. In this architecture, problems related to camera initialization and calibration, energy savings and reliability are dealt with. Camera initialization and calibration are fundamental to achieving a robust identification and classification of the objects present in the scene. Likewise, a system made up of two cameras, where there is cooperation between a static camera with a wide field of view and a rotating camera (pan-tilt-zoom), which allows one to obtain higher resolution images, can be considered a visual multisensory system. The goal is, when the static camera detects an object, for the rotating camera to turn and zoom in so that the object is within its field of view. This type of cooperation is proposed in, for example, Trivedi *et al.* (2002), Costello *et al.* (2004) and Horaud *et al.* (2006) for traffic scenes, to observe people in the observed scenario and to track someone walking or riding a bicycle, respectively. Nguyen *et al.* (2003) propose an architecture to track people moving inside a building. In this architecture, several static cameras are used with overlapping in the field of view allowed. Each camera is coupled to a camera processing module (CPM), which is running on the computer that the camera is connected to. A central module keeps the set of objects currently in the scene and coordinates the tracking tasks between CPMs. In addition, an algorithm is proposed to determine the most appropriate camera to track an object.

In Table 1, the characteristics of the systems described in this section are summed up. Basically, we mention the type of sensors used, the type of architecture, the application environment (indoor or outdoor) and other characteristics are also highlighted. In summary, it has been observed that the architecture of a surveillance system depends greatly on the application. Generally, systems are designed to be used within a specific context and cannot adapt easily to be used in other contexts.

Table 1 Summary of multisensory distributed surveillance systems not using agent technology

Reference	Sensors	Description
Trivedi *et al.* (2000)	Rectilinear, omnidirectional, infrared, range cameras, microphones	Application to traffic incident detection and management. Sensory information is available over wireless networks. Sensor clusters can communicate with each other. Application to outdoor environments.
Velastin *et al.* (2004)	CCTV, local camera network, smart cards and audio sensors	Development for public transport applications. It incorporates functionalities to detect abnormal events, such as standing, intrusion, queue lengths, counter-flow pedestrian traffic and people distribution. Sensors are geographically distributed but in a centralized architecture. CORBA-based communications. Wireless data transmission. Application to indoor environments.
Petrushin *et al.* (2006)	Web cams, an infrared badge ID system, a PTZ camera, a fingerprint reader	Application for locating and tracking people in an office environment. It uses a Bayesian framework to carry out the reasoning. There are cameras with overlapped fields of vision. It uses color histograms in different color spaces (RGB, normalized RGB, HSV) to distinguish people based on their clothes. Application to indoor environments.
Räty *et al.* (2008)	Fingerprint, camera, audio, network analyzing monitor	Application for monitoring just one location. Centralized architecture in a session server. It sends results to the human surveillance operator and the nomadic guard. Application to indoor environments.
Kulkarni *et al.* (2005)	Cyclops, CMUcam, web cams and PTZ cameras	Visual sensor hierarchically organized in three levels. Surveillance tasks (object detection, localization and tracking) are carried out in different levels. The architecture seeks to provide a low-latency yet energy-efficient camera sensing solution. It uses wireless networks. It provides overlapping capacities among cameras.
Trivedi *et al.* (2002)	PTZ, static and mobile omnidirectional, infrared cameras	System for traffic and incident monitoring. There are some overlapped cameras. Application to outdoor environments.
Costello *et al.* (2004)	PTZ cameras	Scheduling policies to observe multiple people using active cameras are described. It is capable of classify a moving object as person, group of people or vehicle. Experiments are simulated.
Horaud *et al.* (2006)	Static, rotating cameras	Computational model for visual attention using cooperation between two cameras. The goal is to control a camera to gaze in a selected direction. It has been tested to follow a pedestrian and a bicycle rider in an outdoor environment.
Nguyen *et al.* (2003)	Static cameras	Application for people tracking in an indoor environment. Centralized architecture. It is possible to have overlapped cameras.

CCTV = closed-circuit television; PTZ = pan-tilt-zoom; CMUcam = low cost, low power sensor created at CMU (Carnegie Mellon University).

According to Attwood and Watson (2004), it is impossible to define one single general purpose architecture optimum to carry out intelligent surveillance, since there are too many variables and restrictions which vary depending on the specific installation and the user's requirements.

The systems previously described do not use agents, when the use of a multiagent distribution approximation can, as a matter of fact, offer several advantages for developing a surveillance system (Valera & Velastin, 2006). First, intelligent cooperation between agents may allow the use of less expensive sensors, and therefore a larger number of sensors may be deployed over a greater area. Second, robustness is increased since, even if some agents fail, others remain to perform the mission. Third, performance is more flexible and there is a distribution of tasks at various locations between groups of agents.

4 Agent-based technology in surveillance-related topics

The general problem which a surveillance system faces can always be broken down into subproblems. Generally, a surveillance system must detect the presence of objects which move in its field of view, classify them into several categories and track them throughout time. Moreover, it must also be able to generate a description of the events which take place in order to interpret the activities that the objects are performing.

Next, the current state of the agent-based technology in surveillance-related topics is offered. First, the use of agents at the three different general abstraction levels in a surveillance system (segmentation, tracking and situation detection) is shown. Next, three typical surveillance video applications, such as traffic surveillance, facial identification and mobile video surveillance by means of mobile platforms for which agents lately have largely contributed, are introduced. We also emphasize the importance of agent-related technology in the topic of multisensory data fusion in surveillance.

4.1. Use of agents in segmentation

Image segmentation is a process which extracts the objects of interest inserted into a captured scene. It consists of allocating pixels from the image to disjoint and homogeneous subsets which configure a partition of the image. Segmentation is the first phase in video processing. Therefore, it is of vital importance to get good results in higher level problems (classification, tracking and activity interpretation). According to Bovenkamp *et al.* (2004), using an MAS for image segmentation has several advantages: (1) it is possible to separate knowledge representation of different domains (problem solving, image processing and application domain); (2) it is possible to separate the image processing algorithms from the control strategies; (3) it is easy to construct and maintain; (4) it provides the ability to benefit from the parallel architectures; (5) it enables one to focus on the ability—all knowledge is not necessary for all tasks; (6) it provides solution for heterogeneous problems; and (7) it contributes to greater reliability—an agent can be corrected by others.

Liu and Tang (1999) propose a system based on reactive agents for brain MRI segmentation. The agents have four reactive behaviors: breeding, diffusion, pixel labeling and decay. In Richard *et al.* (2004), this same problem is dealt with by proposing a hierarchical architecture of situated and cooperative agents. Three types of agents are used: global control agents, local control agents and tissue-dedicated agents. The role of the global control agent is to partition the volume of data into adjacent territories and assign a local control agent to each territory. The role of a local control agent is to create tissue-dedicated agents which implement a local region growing. In Rodin *et al.* (2004), an MAS made up of reactive agents is introduced to detect borders in biological images. It uses the basic representation of interactive components formalism for MAS modeling. The agent behaviors are implemented using the oRis language. The MAS detects concentric striae using two types of agents called darkening and lightening agents to follow the dark and light rings, respectively, which act on the image. In Bovenkamp *et al.* (2004), the researchers basing themselves on the cognitive architecture soar to develop an application which enables intravascular ultrasound image segmentation using agents. Agent knowledge model is divided into modules of knowledge rules which can be added or eliminated, independently, for every agent. This article does not propose any agent or behavior to face the problem of uncertainty and of noisy data. The four systems, previously mentioned, follow a supervised approach. Their goal is image segmentation in regions previously known and expected. However, the system proposed in Mazouzi *et al.* (2007) tries to be general, not supervised. It uses a set of reactive autonomous agents to segment a range image (defined by a regular array of values that describe a surface) in its different planar regions.

4.2 Use of agents for tracking

The agent paradigm has also been used to carry out object tracking with the collaboration of several cameras. Ukita and Matsuyama (2005) propose a three-layer architecture to track several target objects in real time. The system is made up of a group of active vision agents (AVAs). Each AVA is associated with an active camera (fixed-viewpoint pan-tilt-zoom camera) and it can track one single object at a time instant. AVAs that track the same target object form an agency. The architecture consists of three layers, namely, intra-AVA, intra-agency and inter-agency. The configuration of the agencies varies in execution time regarding these parameters: task-constraint, object-priority and utility-function. As far as camera configuration is concerned, two restrictions are necessary: (1) overlapping camera field of views and (2) all observation spaces can be observed by at least two cameras to reconstruct an object's 3D information. In Kim *et al.* (2006), a system made up of several camera agents and a support module to extract object positioning by means of collaboration from several active cameras arranged in a complex environment is presented. Each camera agent is in charge of processing images, obtaining useful information for tracking and controlling the PTZ functions of the linked camera. The support module is in charge of managing the communication between camera agents. In contrast to the previous system, this system does not have the capability to track several objects concurrently. In Garcia *et al.* (2005), the images captured by a camera are supposed to have a central area for optimal vision where images captured by different cameras can overlap. When a target object does not fall within the central area, it must be assigned to a different camera or the observing camera must be centered. On the basis of this configuration, coordination strategies are planned between camera agents to monitor a target object, which has been recently detected, reassign the target object to a different camera or cover hidden areas every so often.

Qureshi and Terzopoulos (2006) use the virtual reality paradigm to experiment and propose a control strategy in order for several cameras to collaborate in persistent human surveillance moving along their fields of view. The surveillance network is made up of non-calibrated static and PTZ cameras. A central server acquires information from the environment through static cameras. This information is used for scheduling active (PTZ) cameras in order to visually examine each person in the scene. PTZ cameras are modeled as autonomous agents capable of recording a video of the person of interest without needing continuous feedback from the central controller. However, Aguilar-Ponce *et al.* (2007) propose a distributed surveillance architecture to detect and track objects of interest. Each camera is controlled by an object processing unit (OPU), which uses an agent-based approach to detect and track the moving objects in the scene. In the OPU, a region agent manages the creation of object agents. A group of OPUs forms a cluster. One OPU plays the role of cluster head and is in charge of merging data sent by other members of the cluster, before sending them to the Scene Processing Unit (SPU), thus saving bandwidth and eliminating redundant data. The SPU will analyze the information it receives to detect threat patterns and carry out the appropriate action. The agents utilize a protocol based on the KQML (knowledge query manipulation language), a popular language widely used for communication among agents (Finin *et al.*, 1992).

Finally, a multiagent framework to manage visual sensor networks is shown in Patricio *et al.* (2008). It includes six types of agents: (1) Each camera is represented and managed by an individual software agent, called the surveillance sensor agent (SSA). Each SSA has partial knowledge due to its limited field of view. It makes decisions with this limitation. Each SSA tracks every target object moving in its local field of view. (2) Cameras with recording functions also have a recording agent assigned to them. (3) The fusion agent integrates the data sent by the assigned SSA agents. It also analyzes the situation to manage the resources and coordinate the associated SSA during the fusion stage. (4) A planning agent will provide a scene overview and make inferences about the target objects and the situation. (5) A context agent provides context-aware information of the scene. (6) An interface agent will provide a graphic user interface to the end user. The belief–desire–intention (BDI) model is used to implement the deliberation and reasoning done over the images captured by the camera, and the communication language. FIPA-ACL is the communication language between the agents. Notice that FIPA stands for Foundation for Intelligent Physical Agents and ACL means agent-communication language.

A mobile agent is a piece of software which is able to migrate from one computer to another. Its use has several advantages (Lange & Oshima, 1999): (1) it reduces network load; (2) network latency is not a problem since interaction can be carried out locally; (3) they can encapsulate protocols; (4) they execute autonomously and asynchronously; (5) they have the capability of adapting dynamically; (6) they are usually computer- and transport-layer-independent, and therefore they can easily integrate into heterogeneous hardware and software systems; and (7) they are robust and fault-tolerant. Likewise, there are disadvantages (Vigna, 2004). Some examples are of the existence of situations in which they are not appropriate, such as difficulties in designing, developing, testing, debugging, authenticating and controlling mobile agents. However, some tasks have taken advantage of the agent-mobility feature to perform surveillance tasks with smart cameras.

In Bramberger *et al.* (2005), mobile agents are used to implement a multicamera tracking method on embedded smart cameras. The tracking algorithm used is KLT (the Kanade–Lucas–Tomasi feature tracker). The physically adjacent (co-located) cameras are grouped into logical clusters named ‘surveillance clusters’. The set of surveillance tasks is distributed among the surveillance tasks and can only migrate to cameras which belong to the same cluster. A camera may be a member of several surveillance clusters. The previous work is improved in Quaritsch *et al.* (2007). The goal is to initiate a single tracking instance for each object of interest. The tracker follows the observed object through the camera network, migrating to the observing camera. In this case, the tracking algorithm used is CamShift (continuously adaptive mean-shift). This solution is autonomous and scalable since it does not require centralized coordination. It is also appropriate for configuration of few cameras with little or no overlapping fields of view. Experiments have been carried out with two smart cameras. There are two types of agents in the cameras: digital signal processor (DSP) agents and smart camera (SmartCam) agents. DSP agents are used to represent video processing tasks. However, SmartCam agents do not interact with DSP agents. They usually carry out control and management tasks. There is an agency in each camera which provides the environment for the mobile agents. Besides, the agency also has a set of system agents which offers services to the DSP and Smart Cam agents. The DIET multiagent platform (`http://diet-agents.sf.net`) has been used to implement the prototype. Moreover, in Jovanovic and Rinner (2007), mobile agents are used to implement a middleware which enables the dynamic reconfiguration of a smart camera system. This permits the modification, in execution time, of the functionality of the camera network. In this work, the exchange of software tasks, as well as the modification of the levels of quality of service provided by the said tasks, is contemplated.

4.3 Use of agents in detection of activities

Interpreting the activities carried out by the objects is a higher level surveillance task than those previously mentioned (segmentation and tracking). In the literature, we can find several works focused on the use of agents to detect situations which are taking place in the observed scenario.

Remagnino *et al.* (1998) propose a surveillance system based on agents to monitor scenes with traffic and pedestrians. There is a textual description of the activity taking place in the 3D world. The annotation of the scene is carried out at two levels: object and situation. As far as objects are concerned, each object tracked (pedestrian or vehicle) is assigned to a behavior agent, which uses a Bayesian network to infer the fundamental characteristics of the object’s trajectory and continuously updates its description. The Bayesian network includes concepts such as object location, object speed, curvature of the most recently analyzed stretch of trajectory, etc. Interaction between objects is interpreted by a situation agent, created dynamically when two objects are near. A threshold (4 m) is used over the Euclidean distance between two objects. This agent uses a Bayesian network to reason about the interaction between objects. Moreover, the authors plan to include an agent, on a level scene, capable of combining situations and behaviors to generate more compact descriptions. Interactions in which more than two objects participate are not considered, although this could be solved by considering interaction approximation between two objects and later the scene agent would combine this information to describe a more global annotation.

Later, in contrast to the previous work where only one camera was considered, a multiagent architecture is proposed to understand the dynamics of the scene based on the information obtained by several cameras, which can have overlapping fields of view (Remagnino *et al.*, 2004). A camera agent detects and tracks all objects in its field of view. It is also in charge of calibration. For each new event (an object entering the camera's field of view), an agent object is created which is in charge of keeping the description of the event updated. Owing to the fact that an event can fall into more than one field of view, object agents, which are identified having been assigned the same object, will merge into one object agent. Once an instance for the object is created, its goals are (1) to classify the event it is associated with, and (2) to classify the behavior based on the limited number of domain-specific activities. The behavior (person leaving the vehicle, vehicle entering, vehicle parking, etc.) is classified based on the information from the object's trajectory. Camera agents with overlapping views create societies of agents with their own agendas. These agendas, which are made by the user or the system administrator, may include the following information: detecting unusual dynamics, keeping the normality of the events in the scene updated, keeping a parameterized log of the events (day of the week and time). It becomes necessary to solve conflicts when there are contradictory hypotheses (e.g. information about velocity and position of an object).

ISIDIS (De Gregorio, 2007) is an active surveillance system for outdoor environments which makes it possible to distinguish between normal and abnormal situations (e.g. a person entering a tunnel, an animal on the railway, an obstacle before the entrance at the tunnel). ISIDIS is made up of three modules: frame elaboration module, virtual neural sensors module and symbolic reasoning module. The elaboration module is in charge of converting color images to binary format. The virtual neural sensors module is made up of two submodules: (1) a setting module makes it possible for the user to place the sensors in the scene (offline) and (2) a sensor reading module receives a black and white image as input and feeds the symbolic reasoning module with a set of responses (sensor values). The symbolic reasoning module, implemented as a BDI agent, interprets and evaluates sensor outputs to determine the alert level. It also decides if the neural network must train or carry out the classifying task. When an abnormal situation is detected, the reasoning module shows the scene and warning level on the monitor of the control center.

The service-oriented computing (SOC) paradigm (Singh & Huhns, 2005) refers to the set of concepts, principles and methods that represent computing in a service-oriented architecture (SOA) in which software applications are constructed based on independent component services with standard interfaces. The main idea of SOC/SOA is to explicitly separate software engineering from programming, to emphasize on software engineering and to de-emphasize on programming. Recently, agents are also being used in conjunction with the SOC paradigm to develop surveillance applications (Vallejo *et al.*, 2008). On the one hand, the services provide the necessary independence and inter-operability levels to tackle system design from a general point of view. On the other hand, agents contribute to providing the necessary autonomy to manage the services. The reasoning kernel is made up of two types of reasoning agents (reasoning agent and security guard) and a reasoning agent factory which offers operations to create, to search and to list the available reasoning agents. The reasoning agent is designed to reason about a certain concept of normality. The definition of normality in an environment is a step toward detecting all situations that are out of normality; that is, abnormal situations. Moreover, the most common abnormal situations can be managed specifically. At any given time, different reasoning agents can coexist, reasoning about different concepts. To analyze normal behavior, general knowledge about the scene, as well as spatial and temporal restrictions and actors coupled to the scene, is used. The reasoning agent also implements a query service to find out if the concept of normality is being given and with which belief. Finally, the security guard agent integrates the knowledge (conclusions) obtained by the reasoning agents; that is, it reasons from a global perspective. Besides, it communicates with the decision-making layer to act in an abnormal situation.

ZeroC Ice (`http:/www.zeroc.com/`), an object-oriented middleware, is used to manage agent communication and facilitate surveillance system deployment. In Detmold *et al.* (2006b), a middleware to support both computation and communication in a surveillance system is shown. Computation is based on a blackboard architectural model organized at three levels: scene analysis level, multiscene analysis level and reasoning level. The communication follows a service-oriented model/architecture. A blackboard is used, instead of a multiagent system, for two reasons (Detmold *et al.*, 2006a): (1) they think that the volume of data managed requires purpose-specific handling; that is, a blackboard, which would be difficult to replicate in an MAS—in a system made up of thousands of cameras, about 26 terabyte (TB) of data are generated daily—and (2) the use of a blackboard is closer to the requirements demanded by the operator, especially when they want to exert greater control than usual. Nevertheless, they indicate that MAS have the advantages of scalability and availability and will therefore conduct research into the application of this paradigm.

4.4 Use of agents for robotic application modeling

In surveillance applications, mobile robots can be sent to hidden, difficult-access or dangerous-for-human personnel areas in the observed scenario in order to generate alert prediagnosis of what is going on in those areas of interest. Over the last few years, numerous methodologies for the development of software systems have emerged which have contributed to the possibility of considering the concept of agent and related terms as a new abstraction, which can be used all along the software life cycle to develop new applications. The main advantage in the use of a methodology is that the knowledge acquired when developing applications can be reused or adapted to new projects. Moreover, using a methodology allows one to share the same terminology, notation, models and development process. However, as far as we know, the only methodologies used to analyze and design robotic systems are Cassiopeia (Collinot *et al.*, 1996), MaSE (DeLoach *et al.*, 2002), PASSI (Cossentino *et al.*, 2003), a methodology which uses concepts from the GAIA, Mas-commonKADS and MaSe methodologies (Jiménez *et al.*, 2007), and recently, Prometheus (Gascueña & Fernández-Caballero, 2009a) has been applied to detect and track a person using a mobile robot. Out of the previously mentioned methodologies, only Cassiopeia and MaSE have been used to design collaborative robotic applications. INGENIAS was followed to model a surveillance environment where there can be different types of entities (e.g. cameras, robots or microphones) and eventually generate JADE code from the proposed model (Pavón *et al.*, 2007). Recently, Tran (2006) has introduced an excellent comparison between the agent-oriented methodologies cited previously.

In Menegatti *et al.* (2007), a distributed perception system made up of an acoustic and a static vision agent, which cooperates with a mobile vision agent mounted on a mobile robot, is presented. The application scenario is in a room environment (one or more) with a dynamic structure; that is, the obstacles that may appear in the environment can change positions. Experiments have been conducted to track an intruder; however, the system is qualified to track several intruders simultaneously. In this work, no agent-oriented methodology was used to model the application.

4.5. Use of agents in traffic surveillance applications

The use of agents is also relevant in traffic surveillance applications. Distributed vehicle monitoring testbed (DVMT) is one of the first applications to enable the testing of many of the concepts in the development of MAS techniques. DVMT is an experiment environment which simulates node networking to carry out a distributed supervision of the vehicles which move among them (Durfee *et al.*, 1989). Spatially distributed nodes detect the sounds of vehicles to track them. Each node is a blackboard-based problem solver, with the appropriate abstraction levels and knowledge sources to carry out vehicle monitoring. This system follows the functionally accurate, cooperative distributed system model. It is characterized by the fact that the agents do not need to have all the information locally to solve their subproblems. To solve problems, they can interact with the rest and exchange partial results. Monitorix (Abreu *et al.*, 2000) is a video-based traffic surveillance MAS where vehicle monitoring is done through a traffic model and some learning algorithms which adjust the parameters of the model. It is multitiered. At the lower level, there are sensors and effectors. High-level semantic information is dealt with in the objective description layer located above the previous layer. There are agents which provide a repository of information, tracking, description of behaviors and classification services. The third layer, the application assistant layer, is where the information from the previous layers is analyzed to take action and make concrete decisions for a specific application. Finally, in the user-assistant layer, the user establishes application profiles which condition the interactions in the lower layers. It uses an FIPA platform and FIPA-ACL for its implementation. Subsequently, Haesevoets *et al.* (2007) propose dynamically forming agent organizations to monitor traffic jams. Each camera has a coupled agent which collaborates with agents from other cameras that detect the same traffic jam.

4.6 Use of agents for facial identification

One of the most useful services in a video surveillance system is the capability to recognize faces. iJADE Face Recognizer is a pose and scale invariant human face recognition MAS (Ao Ieong & Lee, 2004). It is made up of four types of iJADE agents. A face detection agent, located in the client machine, interacts as a user interface and sends all the information extracted on the pose and size of all the face-candidates to the feature extraction modules. A feature selection agent, located in the server, is in charge of selecting the feature extraction method to be used. A face identification agent, located in the center of face recognition, identifies the face. The results obtained are sent to the face detection agent through the messenger agent. System scalability and efficiency can increase by adding more iJADE agents.

In Kipsic and Ribaric (2005), agents are used for the detection and localization of faces in color images. The agents use skin-color features to locate pixels that correspond to the skin. A hybrid-color model is used (HIS-RGB) which is made up of three components (hue, saturation and red). Initially, the agents are uniformly or randomly distributed through the image. They immediately start acting according to their models of behavior (diffusion, breeding or dying) in order to obtain one or more face-candidate regions that represent the face. Finally, a heuristically defined decision function, based on five features (face orientation and size, width/height ratio, regularity and fullness of the face-region candidate), is applied to determine whether the face-candidate region is really a face.

In Shunmuganathan and Vasudevan (2006), RETSINA (Reusable Environment for Task-Structured Intelligent Network Agents) is used to develop a facial recognition MAS. RETSINA is a framework for developing MAS applications, according to FIPA specifications. In the system, each RETSINA agent carries out facial recognition with its own training set. When there is a match with the face being analyzed, an ACL message is sent to the rest of the agents to notify them of the match. At its worst, the agent-based facial recognition system has proven to be 80% better than the ordinary facial recognition system.

Chetty and Sharma (2006) adapt the MAS MARSE framework to implement a distributed facial recognition task. The system architecture is a four-tier structure. In the central controller layer, there are three types of agents. A skin segmentation agent locates the facial region based on the statistical distribution of the skin color and on a threshold. A skin region detectionagent detects the facial region by matching skin regions with a template. A face fusion agent fuses the results obtained based on several color spaces (HSV, YCrCb, RGB). Moreover, in the remote application layer, there is also a matching agent that interacts with remote agents and communicates with the image databases to carry out searches and matches. To implement the system, the IBM Agent Building and Learning Environment (ABLE) is used. ABLE (Bigus, 2000) is a JAVA-developed framework to develop intelligent agents using machine learning and reasoning.

4.7. Use of agents for multisensory data fusion

Obtaining information about the targets moving along the observed scenario is approached, for some surveillance applications, as a multisensory data fusion problem. The data fusion process consists of combining data obtained by the sensors, or their derivatives, in order to obtain information of a greater quality than what would be obtained if the data sources were used individually (Mitchell, 2007).

In Karlsson *et al.* (2005), the agent-oriented programming paradigm is used to develop intelligent sensor networks. JADE is chosen to implement an Unattended Ground Sensor Network (UGSN) to monitor moving targets. Fusion is possible thanks to the collaborative and mobile features of the software agents created. A mobile agent called TargetAgent (TA) is created in the sensor coupled to the host which detects a target. Then, the TA reasons with its neighboring nodes, correlating the data to the sensors, to decide where to migrate next. The TA migrates from node to node to be close to the target it is following at all times. After every migration, the TA carries out a distributed fusion process with the available information from the sensor of the new host. Likewise, in Biswas *et al.* (2008), the use of mobile agents is proposed to carry out collaborative data fusion in sensor networks. The results of the experiment, classifying targets moving in a sensor network, show that mobile agent-based collaborative fusion has advantages over the client-server model due to the amount of sensors deployed and the degree of collaboration needed. The mobile agent-based model tends to use less execution time and energy than the client-server model when the number of nodes is large.

In Castanedo *et al.* (2008), two fusion models are applied (active and passive fusion) to a distributed visual sensors network for tracking purposes. In passive fusion, each surveillance-sensor agent processes the information from its field of view and sends track information to the fusion agent. In active fusion, feedback information is provided to every agent involved in the fusion process. This information allows every agent to reason about the quality of the information it is sending in regard to overlapping sensors. Since each sensor is autonomous, it can decide on the inconsistencies of the information and correct them before sending the information to the fusion agent. The results show that the active fusion model performs better than the passive fusion model. Moreover, in Perugini *et al.* (2003), it is mentioned that agents are appropriate for fusion because they can represent autonomous fusion entities, modeling their capabilities, expertise and intentions. The data fusion agents' application is illustrated at the levels of impact and situation fusion of the λJDL model, a revised model of the JDL data fusion process model. At the situation fusion level, one or more data sources are used over time to gather a representation of relations of interest between objects of interest in an environment. The fusion impact level uses one or more data sources over time to gather a representation of effects of situations in an environment, relative to intentions.

The concept of coalition was used in CSA (Cooperative Sensor Agents) architecture (Castanedo *et al.*, 2006) to describe the collaboration between a group of autonomous agents that decide to work together to reach a temporary common goal. CSA architecture has two levels: the sensor layer and the coalition layer. A coalition is formed when an agent (sensor) needs to cooperate with other agents with capabilities it does not have. The coalition ends when the task ends. The agents who follow a BDI model are implemented with the Jadex language. Agent communication is carried out through FIPA messages. In the MAS, there is at least one agent with tracking capabilities, another with recognition capabilities, a third with calibration capabilities and yet another with recording capabilities.

Sensor managing is another of the important tasks to be carried out in multisensory systems. It consists of optimizing the global management of the sensor system through the application of individual operations in each sensor. In Molina *et al.* (2003), a collaborative assignment of the tasks to be carried out by each sensor (multifunction radar), using agents, is proposed. The fusion agent periodically sends a list with the tasks to be performed by the system to the sensor agents. The sensor agent periodically decides which tasks the sensor must perform. The problem with

Table 2 Summary of the use of agent technology in surveillance systems

Task	Description	References
Activity recognition	The system only informs about the activities that it is able to recognize (e.g. walking through or near the parking lot, reaching or getting out of a vehicle; passing near, through or parking in the parking lot; man crossing the railway, a couple taking a walk on the railway; left luggage)	Remagnino *et al.* (1998, 2004), De Gregorio (2007), Detmold *et al.* (2006a)
	The system identifies as anomalous situations those that are not defined as normal. Pedestrian walking along the pavement or crossing over the pedestrian crossing are some examples of normal situations	Vallejo *et al.* (2008)
Facial recognition	Application of biometric identification through facial recognition	Ao Ieong and Lee (2004), Kipsic and Ribaric (2005), Shunmuganathan and Vasudevan (2006), Chetty and Sharma (2006)
Multisensor data fusion	Mobile agents are used to carry out the multisensor data fusion in a distributed manner	Karlsson *et al.* (2005), Biswas *et al.* (2008)
	Applications that do not use mobile agents in data fusion	Castanedo *et al.* (2008), Perugini *et al.* (2003), Castanedo *et al.* (2006), Molina *et al.* (2003)
Robotic applications	Applications modeled using an agent-oriented software engineering methodology	Collinot *et al.* (1996), DeLoach *et al.* (2002), Cossentino *et al.* (2003), Jiménez *et al.* (2007), Pavón *et al.* (2007), Gascueña and Fernández-Caballero (2009a)
Segmentation	Reactive agent collection specialized in interpreting biological or medical image regions. Unsupervised approach is followed	Liu and Tang (1999), Richard *et al.* (2004), Rodin *et al.* (2004), Bovenkamp *et al.* (2004)
	A supervised approach is followed to interpret image regions	Mazouzi *et al.* (2007)
Tracking	Mobile agents are used in order to carry out tracking tasks. Central control host is not required. Smart cameras are employed	Bramberger *et al.* (2005), Quaritsch *et al.* (2007), Jovanovic and Rinner (2007)
	Systems that do not use mobile agents or smart cameras in tracking tasks	Ukita and Matsuyama (2005), Kim *et al.* (2006), García *et al.* (2005), Qureshi and Terzopoulos (2006), Aguilar-Ponce *et al.* (2007), Patricio *et al.* (2008)
Traffic surveillance	Applications such as vehicle monitoring or detecting traffic jams	Durfee *et al.* (1989), Abreu *et al.* (2000), Haesevoets *et al.* (2007)

assigning tasks to sensors is the result of a negotiation process carried out between the sensor agents, beginning with the information sent to them by the fusion agent (priority and performance objectives). Fuzzy logic is used to evaluate multisensor task priorities. A symbolic bottom-up fuzzy reasoning process is performed, which considers the available fused or local target tracks, surveillance sectors data and (external) intelligence information. As a result of these reasoning processes, performed at each agent planning level, the priorities of system-level and sensor-level tasks will be inferred and applied during the negotiation process.

Table 2 shows a summary of the works that use agent technology in surveillance systems, outlining the main focus of their work, as well as a brief description.

5 Discussion and conclusions

Again, we must outline that current surveillance systems are noted more and more for the use of several devices, heterogeneous in many instances, distributed along the observed scenario. The incorporation of an intelligence layer to alert the operator proactively about what is going on in the observed scenario and prevent the observer from having to observe the monitors continuously is an unavoidable goal nowadays. Hence, now more than ever, basic agent characteristics (autonomy, reactivity, proactiveness and social ability), along with MAS characteristics (distributed data management, low coupling, robustness, communication and coordination between autonomous entities), bring us to the conclusion that the agency is an excellent choice for solving the mind-boggling problems in surveillance systems.

As far as segmentation is concerned, in this study, the use of reactive agents to process images at low levels is reduced to the medical and biological scopes. This technology adapts well to images captured in these scopes due to their characteristics. For instance, tumors, healthy parts, etc. in the brain have very well-defined characteristics. Hence, specific agents in charge of identifying each type of area to be extracted from the image can be defined. However, we have not been able to extrapolate these results for use in surveillance systems components. The reason for this could be that a surveillance environment has certain conditions which can vary dynamically, having to face such complex problems as shadows, ghosts or image reflections, changes in lighting and weather conditions. In such cases, traditional techniques used in computer vision without agents are used.

It makes more sense and more work has been carried out in the use of agents to solve problems dealt with at levels higher than those of segmentation: tracking and activity detection. In the problem of tracking, the use of an agent for every camera installed in the observed environment is a pretty generalized idea. A set of cameras are deployed along the environment. Cooperation between agents allows one to obtain more information about what is going on at the scene than what is obtained with the use of a single camera. The use of the mobility feature of the agents was also exploited to do the tracking. The agent migrates from one camera to another to track the coupled object. In the problem of activity detection, several agents are used, which collaborate by reasoning and offering a description of what is going on. The conceptual idea of leading the performance of the agents based on a BDI deliberative architecture has been put into practice in some surveillance systems at the tracking and activity detection level. This model is one of the most widely used in the agent community and it is based on three concepts very close to human reasoning: beliefs, desires and intentions.

The systems revised are based on the conceptual idea of the agent, but many of them use a non-agent-oriented framework for their implementation instead of an agent programming language (JACK, JADE and Jadex). Therefore, it is difficult to discern the traceability between the concepts used in the conception of the system and those used in its implementation. Agent-oriented software engineering (AOSE) has emerged in response to urgent needs which deal with methods and frameworks for requirements engineering, analysis, design and implementation; languages for programming, communication and coordination and ontology specification; and development tools and platforms (Weiss, 2001). Nevertheless, the problem occurs in many other applications developed according to the agent paradigm, not only in surveillance. Only some applications in the robotics field have been developed following an AOSE methodology. These methodologies provide designers and developers with a way to structure an application based on communication between autonomous elements.

Given the current proliferation of agent-based surveillance applications, as previously seen, we believe much more attention should be given to all aspects related to agent-based methodologies. Likewise, we believe this is evident in the advantages of applying agent-based methodologies in robotics applications, as well as in the advantages of the use of methodologies in surveillance applications in general (Gascueña & Fernández-Caballero, 2007). Indeed, the main advantage in the use of a methodology is that the knowledge acquired when developing applications can be reused or adapted to new projects. Traditionally, surveillance applications have focused directly on proposing algorithms and architectures to carry out surveillance tasks (López *et al.*, 2006b; Fernández-Caballero *et al.*, 2008). However, they have developed without following a strict development process from analysis and application design to code generation. It would be desirable to follow a more engineering approach because a well documented and structured product would be obtained along every developing phase. The use of a methodology would help designers understand and develop the application.

In addition, nowadays the use of MDE (model-driven engineering) techniques along with the software development life cycle is gaining more and more interest (Schmidt, 2006). Model-driven development (MDD) has important benefits in fundamental aspects such as productivity, portability, inter-operability and maintenance. Therefore, in the MAS field, it seems quite useful to use MAS methodologies incorporating the MDD paradigm (Gascueña & Fernández-Caballero, 2009b, 2009c).

Finally, it should be noted that there are still many concepts related to agent technology that have not been exploited in the development of surveillance systems (e.g. auctions, negotiations, FIPA interaction protocols, planning, trust and reputation; Beer *et al.*, 1999). For instance, there is no study of the best suited multiagent system organizational paradigms (e.g. hierarchies, holarchies, coalitions, teams, congregations, societies, federations, markets and matrix organizations) in surveillance systems (Horling & Lesser, 2004).

Acknowledgments

This work was partially supported by the Spanish Ministerio de Ciencia e Innovación under projects TIN2007-67586-C02-02 and TIN2010-20845-C03-01 and by the Spanish Junta de Comunidades de Castilla-La Mancha under projects PII2I09-0069-0994, PII2I09-0071-3947 and PEII09-0054-9581.

References

Abreu, B., Botelho, L., Cavallaro, A., Douxchamps, D., Ebrahimi, T., Figueiredo, P., Macq, B., Mory, B., Nunes, L., Orri, J., Trigueiros, M. J. & Violante, A. 2000. A video-based multiagent traffic surveillance system. In *The IEEE Intelligent Vehicles Symposium*, Dearborn, MI, USA, 457–462.

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

Ao Ieong, T. W. H. & Lee, R. S. T. 2004. iJADE Face Recognizer—A multiagent based pose and scale invariant human face recognition system. In *Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Artificial Intelligence **3214**, 494–601. Springer.

Attwood, C. I. & Watson, D. A. 2004. Advisor—socket and see: lessons learnt in building a real-time distributed surveillance system. In *Intelligent Distributed Surveillance Systems*, London, UK, 6–11.

Beer, M., d'Inverno, M., Luck, M., Jennings, N., Preist, C. & Schroeder, M. 1999. Negotiation in multi-agent systems. *The Knowledge Engineering Review* **14**, 285–289.

Bigus, J. P. 2000. The agent building and learning environment. In *The Fourth International Conference on Autonomous Agents*, Barcelona, Spain, 108–109.

Biswas, P. K., Qi, H. & Xu, Y. 2008. Mobile-agent-based collaborative sensor fusion. *Information Fusion* **9**, 399–411.

Bovenkamp, E. G. P., Dijkstra, J., Bosch, J. G. & Reiber, J. H. C. 2004. Multiagent segmentation of IVUS images. *Pattern Recognition* **37**(4), 647–663.

Bramberger, M., Quaritsch, M., Winkler, T., Rinner, B. & Schwabach, H. 2005. Integrating multicamera tracking into a dynamic task allocation system for smart cameras. In *Advanced Video and Signal Based Surveillance*, Como, Italy, 474–479.

Castanedo, F., Patricio, M. A., García, J. & Molina, J. M. 2006. Extending surveillance systems capabilities using BDI cooperative sensor agents. In *4th ACM International Workshop on Video Surveillance and Sensor Networks*, Santa Barbara, CA, USA, 131–138.

Castanedo, F., García, J., Patricio, M. A. & Molina, J. M. 2008. A multiagent architecture to support active fusion in a visual sensor network. In *2nd ACM/IEE International Conference on Distributed Smart Cameras*, Stanford, CA, USA, 1–8.

Chetty, G. & Sharma, D. 2006. Distributed face recognition: A multiagent approach. In *Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Artificial Intelligence **4253**, 1168–1175. Springer.

Collinot, A., Drogoul, A. & Benhamou, P. 1996. Agent oriented design of a soccer robot team. In *2nd International Conference on Multiagent Systems*, Kyoto, Japan, 41–47.

Cossentino, M., Sabatucci, L. & Chella, A. 2003. A possible approach to the development of robotic multiagent systems. In *IEEE/WIC Conference on Intelligent Agent Technology*, Halifax, Canada, 539–544.

Costello, C., Diehl, C., Banerjee, A. & Fisher, H. 2004. Scheduling an active camera to observe people. In *ACM 2nd International Workshop on Visual Surveillance and Sensor Networks*, New York, NY, USA, 39–45.

Cucchiara, R. 2005. Multimedia surveillance systems. In *The third ACM International Workshop on Video Surveillance and Sensor Networks*, Singapore, 3–10.

De Gregorio, M. 2007. A hybrid intelligent system for active video surveillance. In *The Seventh International Conference on Intelligent Systems Design and Applications*, 21–26.

DeLoach, S., Matson, E. & Li, Y. 2002. Applying agent oriented software engineering to cooperative robotics. In *The Fifteenth International Florida Artificial Intelligence Research Society Conference*, 391–396.

Detmold, H., Dick, A., Falkner, K., Munro, D. S., Hengel, A. & Morrison, R. 2006a. Scalable surveillance software architecture. In *IEEE International Conference on Video and Signal Based Surveillance*, Sydney, Australia, 103–108.

Detmold, H., Dick, A., Falkner, K., Munro, D. S., Hengel, A. & Morrison, R. 2006b. Middleware for video surveillance networks. In *International Workshop on Middleware for Sensor Networks*, Melbourne, Australia, 31–36.

Durfee, E. H., Lesser, V. R. & Corkill, D. D. 1989. Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering* **1**(1), 63–83.

d'Inverno, M., Fisher, M., Lomuscio, A., Luck, M., de Rijke, M., Ryan, M. & Wooldridge, M. 1997. Formalisms for multi-agent systems. *The Knowledge Engineering Review* **12**, 315–321.

Fernández-Caballero, A., Gómez, F. & López-López, J. 2008. Road-traffic monitoring by knowledge-driven static and dynamic image analysis. *Expert Systems with Applications* **353**, 701–719.

Finin, T., McKay, D. & Fritzon, R. 1992. *An Overview on KQML: A Knowledge Query and Manipulation Language*. Technical Report no. CS-94-12, University of Maryland Computer Science Department.

Franklin, S. & Graesser, A. 1996. Is it an agent, or just a program?: a taxonomy for autonomous agents. In *Intelligent Agents III, Agent Theories, Architectures and Languages*, Lecture Notes in Computer Science **1193**, 21–35. Springer.

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

Gascueña, J. M. & Fernández-Caballero, A. 2007. The INGENIAS methodology for advanced surveillance systems modelling. In *2nd International Work-Conference on the Interplay between Natural and Artificial Computation*, Lecture Notes in Computer Science **4528**, 541–550.

Gascueña, J. M. & Fernández-Caballero, A. 2009a. Agent-based modeling of a mobile robot to detect and follow humans. In *3rd International KES Symposium on Agents and Multiagent Systems—Technologies and Applications*, Lecture Notes in Computer Science **5559**, 80–89.

Gascueña, J. M. & Fernández-Caballero, A. 2009b. Prometheus and INGENIAS agent methodologies: a complementary approach. In *Agent-Oriented Software Engineering IX*, Lecture Notes in Computer Science **5386**, 131–144.

Gascueña, J. M. & Fernández-Caballero, A. 2009c. Towards an integrative methodology to develop multiagent systems. In *International Conference on Agents and Artificial Intelligence*, Porto, Portugal, 392–399.

Haering, N., Venetianer, P. L. & Lipton, A. 2008. The evolution of video surveillance: an overview. *Machine Vision and Applications* **19**(5–6), 279–290.

Haesevoets, R., Van Eylen, B., Weyns, D., Helleboogh, A. & Holvoet, T. 2007. Context-driven dynamic organizations applied to coordinated monitoring of traffic jams. In *Engineering Environment-Mediated Multiagent Systems*, Dresden, Germany, 126–143.

Horaud, R. P., Knossow, D. & Michaelis, M. 2006. Camera cooperation for achieving visual attention. *Machine Vision and Applications* **16**(6), 331–342.

Horling, B. & Lesser, V. 2004. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review* **19**, 281–316.

Hu, W., Tan, T., Wang, L. & Maybank, S. 2004. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics* **34**(3), 334–350.

Huhns, M. N. & Stephens, L. M. 1999. Multiagent systems and societies of agents. In *Multiagent Systems*, Weiss, G. (ed.). The MIT Press.

Jiménez, J. A., Vallejo, M. & Ochoa, J. F. 2007. Methodology for the analysis and design of multiagent robotic systems: MAD-Smart. *Avances en Sistemas e Informática* **4**(2), 61–70.

Jovanovic, M. & Rinner, B. 2007. Middleware for dynamic reconfiguration in distributed camera systems. In *5th IEEE International Workshop on Intelligent Solutions in Embedded Systems*, Madrid, Spain, 139–150.

Kang, W. & Deng, F. 2007. Research on intelligent visual surveillance for public security. In *6th IEEE/ACIS International Conference on Computer and Information Science*, Melbourne, Australia, 824–829.

Karlsson, B., Bäckström, O., Kulesza, W. & Axelsson, L. 2005. Intelligent sensor networks—an agent-oriented approach. In *Workshop on Real-World Wireless Sensor Networks*, Stockholm, Sweden.

Kim, N., Kim, I. & Kim, H. 2006. Video surveillance using dynamic configuration of multiple active cameras. In *International Conference on Image Processing*, Atlanta, GA, USA, 1761–1764.

Kipsic, D. & Ribaric, S. 2005. A multiagent-based approach to face detection and localization. In *27th International Conference on Information Technology Interfaces*, Cavtat, Croatia, 377–382.

Kulkarni, P., Ganesan, D., Shenoy, P. & Lu, Q. 2005. SensEye: a multitier camera sensor network. In *13th ACM International Conference on Multimedia*, Singapore, 229–238.

Kumar, P., Mittal, A. & Kumar, P. 2008. Study of robust and intelligent surveillance in visible and multimodal framework. *Informatica* **32**(1), 63–77.

Lange, D. B. & Oshima, M. 1999. Seven good reasons for mobile agents. *Communications of the ACM* **42**(3), 88–89.

Liu, J. & Tang, Y. Y. 1999. Adaptive image segmentation with distributed behaviour based agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(6), 544–551.

López, M. T., Fernández-Caballero, A., Fernández, M. A., Mira, J. & Delgado, A. E. 2006a. Visual surveillance by dynamic visual attention method. *Pattern Recognition* **39**(11), 2194–2211.

López, M. T., Fernández-Caballero, A., Fernández, M. A., Mira, J. & Delgado, A. E. 2006b. Motion features to enhance scene segmentation in active visual attention. *Pattern Recognition Letters* **275**, 469–478.

López, M. T., Fernández-Caballero, A., Fernández, M. A., Mira, J. & Delgado, A. E. 2007. Dynamic visual attention model in image sequences. *Image and Vision Computing* **25**(5), 597–613.

Mazouzi, S., Guessoum, Z., Michel, F. & Batouche, M. 2007. A multiagent approach for range image segmentation. In *5th International Central and Eastern European Conference on MultiAgent Systems*, Leipzig, Germany, 1–10.

Menegatti, E., Cavasin, M., Mumolo, E., Nolich, M. & Pagello, E. 2007. Combining audio and video surveillance with a mobile robot. *International Journal on Artificial Intelligence Tools* **16**(2), 377–398.

Mitchell, H. B. 2007. *MultiSensor Data Fusion: An Introduction*. Springer-Verlag.

Molina, J. M., Garcia, J., Jiménez, F. J. & Casar, J. R. 2003. Cooperative management in a net of intelligent surveillance agent-sensors. *International Journal of Intelligent Sytems* **18**(3), 279–307.

Nguyen, N., Bui, H., Venkatesh, S. & West, G. 2003. Multiple camera coordination in a surveillance system. *ACTA Automatica Sinica* **29**(3), 408–422.

Patricio, M. A., Castanedo, F., Berlanga, A., Pérez, O., García, J. & Molina, J. M. 2008. Computational intelligence in visual sensor networks: improving video processing systems. *Studies in Computational Intelligence* **96**, 351–377.

Pavón, J., Gómez-Sanz, J., Fernández-Caballero, A. & Valencia, J. J. 2007. Development of intelligent multisensor surveillance systems with agents. *Robotics and Autonomous Systems* **55**(12), 892–903.

Pechoucek, M., Thompson, S. G. & Voos, H. (eds). 2008. Defense industry applications of autonomous agents and multiagent systems. In *Whitestein Series in Software Agent Technologies and Autonomic Computing*, Whitestone Technologies.

Perugini, D., Lambert, D., Sterling, L. & Pearce, A. 2003. Distributed information fusion agents. In *Sixth International Conference of Information Fusion*, Cairns, Australia, **1**, 86–93.

Petrushin, V. A., Wei, G., Shakil, O., Roqueiro, D. & Gershman, A. V. 2006. Multiple-sensor indoor surveillance system. In *The 3rd Canadian Conference on Computer and Robot Vision*, Quebec City, Canada, 40.

Quaritsch, M., Kreuzthaler, M., Rinner, B., Bischof, H. & Strobl, B. 2007. Autonomous multicamera tracking on embedded smart cameras. *EURASIP Journal on Embedded Systems*, **2007**(1), 10.

Qureshi, F. & Terzopoulos, D. 2006. Surveillance camera scheduling: a virtual vision approach. *ACM Multimedia Systems Journal* **12**(3), 269–283.

Räty, T., Lindholm, M., Nieminen, M. & Oikarinen, J. 2008. Distributing essential logical deductions to surveillance personnel and a video recorder. In *Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, Valencia, Spain, 295–304.

Remagnino, P., Tan, T. & Baker, K. 1998. Agent orientated annotation in model based visual surveillance. In *Sixth International Conference on Computer Vision*, Bombay, India, 857–862.

Remagnino, P., Shihab, A. I. & Jones, G. A. 2004. Distributed intelligence for multicamera visual surveillance. *Pattern Recognition* **37**(4), 675–689.

Richard, N., Dojat, M. & Garbay, C. 2004. Automated segmentation of human brain MR images using a multiagent approach. *Artificial Intelligence in Medicine* **30**(2), 153–176.

Rodin, V., Benzinou, A., Guillaud, A., Ballet, P., Harrouet, F., Tisseau, J. & Le Bihan, J. 2004. An immune oriented multiagent system for biological image processing. *Pattern Recognition* **37**(4), 631–645.

Schmidt, D. C. 2006. Guest editor's introduction: model-driven engineering. *Computer* **39**(2), 25–31.

Shunmuganathan, K. L. & Vasudevan, V. 2006. Multiagent based facial recognition system using RETSINA. *Asian Journal of Information Technology* **5**(11), 1177–1179.

Singh, M. P. & Huhns, M. N. 2005. *Service-oriented Computing: Semantics, Processes, Agents*. John Wiley & Sons, Ltd.

Tran, Q. N. 2006. *MOBMAS—A Methodology for Ontology-based Multi-agent Systems Development*. PhD dissertation, School of Information Systems, Technology and Management, University of New South Wales.

Trivedi, M., Mikic, I. & Kogut, G. 2000. Distributed video networks for incident detection and management. In *Proceedings of IEEE Conference on Intelligent Transportation Systems*, Dearborn, MI, USA, 155–160.

Trivedi, M., Prati, A. & Kogut, G. 2002. Distributed interactive video arrays for event based analysis of incidents. In *The 5th IEEE Conference on Intelligent Transportation Systems*, Dearborn, MI, USA, 950–956.

Ukita, N. & Matsuyama, T. 2005. Real-time cooperative multitarget tracking by communicating active vision agents. *Computer Vision and Image Understanding* **97**(2), 137–179.

Valera, M. & Velastin, S. A. 2006. A review of the state-of-the-art in distributed surveillance systems. In *Intelligent Distributed Video Surveillance Systems*, The Institution of Engineering and Technology, 1–30.

Vallejo, D., Albusac, J., Gonzalez-Morcillo, C. & Jiménez, L. 2008. A service-oriented multiagent architecture for cognitive surveillance. In *Cooperative Information Agents XII*, Lecture Notes in Computer Science **5180**, 101–115. Springer.

Velastin, S. A., Khoudour, L., Lo, B. P. L., Sun, J. & Vicencio-Silva, M. 2004. PRISMATICA: a multisensor surveillance system for public transport networks. In *The Twelfth International Conference Road Transport Information and Control*, London, UK, 19–25.

Vigna, G. 2004. Mobile agents: ten reasons for failure. In *IEEE International Conference on Mobile Data Management*, Berkeley, CA, USA, 298–299.

Weiss, G. 2001. Agent orientation in software engineering. *The Knowledge Engineering Review* **16**(4), 349–373.

Wooldridge, M. & Jennings, J. A. 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review* **10**(2), 115–152.