Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 98639, 21 pages
doi:10.1155/2007/98639

Research Article

Multi-Agent Framework in Visual Sensor Networks

M. A. Patricio, J. Carbó, O. Pérez, J. García, and J. M. Molina

Grupo de Inteligencia Artificial Aplicada, Departamento de Informática, Universidad Carlos III de Madrid,
Avda. Universidad Carlos III 22, Colmenarejo, Madrid 28270, Spain

Received 4 January 2006; Revised 13 June 2006; Accepted 13 August 2006

Recommended by Ching-Yung Lin

The recent interest in the surveillance of public, military, and commercial scenarios is increasing the need to develop and deploy intelligent and/or automated distributed visual surveillance systems. Many applications based on distributed resources use the so-called software agent technology. In this paper, a multi-agent framework is applied to coordinate videocamera-based surveillance. The ability to coordinate agents improves the global image and task distribution efficiency. In our proposal, a software agent is embedded in each camera and controls the capture parameters. Then coordination is based on the exchange of high-level messages among agents. Agents use an internal symbolic model to interpret the current situation from the messages from all other agents to improve global coordination.

1. INTRODUCTION

Nowadays, surveillance camera systems are applied in transport applications, such as airports [1, 2], sea environments [3, 4], railways, underground [5–9], and motorways to observe traffic [10–14] in public places, such as banks, supermarkets, homes, department stores [15–19], and parking lots [20–22] and in the remote surveillance of human activities such as football match attendance [23] or other activities [24–26]. The common processing tasks that commercial systems perform are intrusion and motion detection [27–32] and packages detection [28, 31, 32]. Research in university groups tends to improve image processing tasks by generating more accurate and robust algorithms for object detection and recognition [22, 33–37], tracking [22, 26, 33, 38–41], human activity recognition [42–44], database [45–47], and tracking performance evaluation tools [48].

Third-generation surveillance systems [49] is the term sometimes used in the literature to refer to systems conceived to deal with a large number of cameras, a geographical spread of resources, many monitoring points, as well as to mirror the hierarchical and distributed nature of the human process of surveillance. From an image processing point of view, they are based on the distribution of processing capacities over the network and the use of embedded signal-processing devices to get the benefits of scalability and potential robustness provided by distributed systems. The main goals that are expected of a generic third-generation vision surveillance application, based on end-user requirements, are that it should provide good scene understanding, aimed at attracting the attention of the human operator in real time, possibly in a multisensor environment, as well as surveillance information using low-cost standard components.

We have developed a novel framework for deliberative camera-agents forming a visual sensor network. This work follows on from previous research on computer vision, information fusion, and intelligent agents. Intelligence in artificial vision systems, such as our proposed framework, operates at different logical levels. First, the process of scene interpretation from each sensor is enacted by an agent-camera. As a second step, the information parsed by a separate local processor is collected and fused. Finally, the surveillance process is distributed over several agent-cameras, according to their individual ability to contribute their local information to a global target solution.

A distributed solution is an option for the problem of coordinating multi-camera systems. It has the advantages of scalability and fault-tolerance over centralization. In our approach, distribution is achieved by a multi-agent system, where each camera is represented and managed by an individual software agent. Each agent knows only part of the information (partial knowledge due to its limited field of view), and has to make decisions with this limitation. The distributedness of this type of systems supports the camera-agents' proactivity, and the cooperation required among these agents to accomplish surveillance justifies the sociability of camera-agents. The intelligence produced by the symbolic internal model of camera-agents is based on a deliberation about the state of the outside world (including its past evolution), and the actions that may take place in the future. Several architectures inspired by different disciplines, like psychology, philosophy, and biology, can be applied to build agents with the ability to deliberate. Most of them are based on theories for describing the behavior of individuals. They include the belief-desire-intention (BDI) model, the theory of agent-oriented programming [50], the unified theories of cognition [51], and subsumption theory [52]. Each of these theories has its strengths and weaknesses and is especially suited for particular kinds of application domains. Of these theories, we have chosen the BDI model to implement the deliberation about the images captured by the camera. Agents sociability presumes some kind of communication between agents. The most accepted agent communication schemes are those based on speech-act Theory (e.g., KQML and FIPA-ACL) [53].

The foundation for most implemented BDI systems is the abstract interpreter proposed by Rao and Georgeff [54]. Although many ad hoc implementations of this interpreter have been applied to several domains, the release of JADEX [55] is gaining acceptance recently. JADEX is an extension of JADE [56], which facilitates FIPA communications between agents, and it is widely used to implement intelligent and software agents. But JADEX also provides a BDI interpreter for the construction of agents. The beliefs, desires, and intentions of JADEX agents are defined easily in XML and Java, enabling researchers to quickly exploit the potential of the BDI model. It is a promising technology that is likely to soon become an unofficial standard for building deliberative agents. Therefore, this was the technology that we chose to implement our multi-agent framework.

The purpose of this paper is to show our multi-agent framework for visual sensor networks applied to surveillance system environments. Visual sensor networks are composed of different sensors that monitor an extended area. The main issue for analyzing information in this distributed environment is to progressively reduce redundancy and coherently combine information and processing capability. In our framework, these objectives are achieved thanks to its coordination abilities, which allow a dynamic distribution of surveillance tasks among the nodes, taking into account their internal state and situation. Two types of scenarios—indoor and outdoor configurations for intrusion detection and tracking—are presented to illustrate this framework's capability to improve the surveillance globally provided by the network. Both scenarios highlight how coordinated operation enhances surveillance systems. The first scenario is related to the robustness and reliability of surveillance output, assessed with special-purpose metrics. On the other hand, the second shows how this framework extends the network functionalities, allowing surveillance tasks to be accomplished automatically, while the cameras are accessible at the same time for human operators. Both scenarios are implemented using the same BDI architecture that is presented in Section 4. Obviously, the only things to be changed are the current state of the world according to each camera-agent's perception, tailored to the specific situation of each scenario. This is a very important feature in surveillance systems, since we usually manage a sizeable number of visual sensors. As we have used the standard representation of a generic camera-agent using JADEX, our framework has the advantage of developing distributed surveillance systems easily.

The remainder of the paper describes our multi-agent framework applied to building distributed visual sensor networks for surveillance. First, Section 2 is a survey of current distributed camera surveillance systems. Section 3 describes the architecture of our framework and details the structure of the agent-cameras represented in terms of the BDI model. Section 4 deals with the problem of managing information in a visual sensor network and the information exchange process between neighboring camera-agents in order to achieve a robust and reliable global surveillance task. Then, two scenarios are presented in Section 5. This section shows the improvements achieved by using this framework and analyzes the gain over situations where there is no coordination at all between visual sensors. Finally, the conclusions are set out in Section 6.

2. DISTRIBUTED CAMERA SURVEILLANCE SYSTEMS: A SURVEY

A typical configuration of processing modules in a camera surveillance system is composed of several stages (Figure 1).

(1) Object detection module. There are two main conventional approaches to object detection: "temporal difference" and "background subtraction." The first approach consists of the subtraction of two consecutive frames followed by thresholding. The second technique is based on the subtraction of a background or reference model and the current image followed by a labelling process. After applying either of these approaches, morphological operations are typically applied to reduce the noise of the image difference.

(2) Object recognition module. This module uses model-based approaches to create constraints in the object appearance model, for example, the constraint that people appear upright and in contact with the ground. The object recognition task then becomes a process of using model-based techniques in an attempt to exploit this knowledge.

(3) A tracking system. A filtering mechanism to predict each movement of the recognized object is a common tracking method. The filter most commonly used in surveillance systems is the Kalman filter [38, 57]. Fitting bounding boxes or ellipses, which are commonly called "blobs," to image regions of maximum probability is another tracking approach based on statistical models. The assumptions made to apply linear or Gaussian filters do not hold in some situations of interest, and then nonlinear Bayesian filters, such as extended Kalman filters (EKF) or particle filters, have been proposed. HMMs (hidden Markov models) are applied for tracking purposes as presented in [58]. Recent research is focusing on

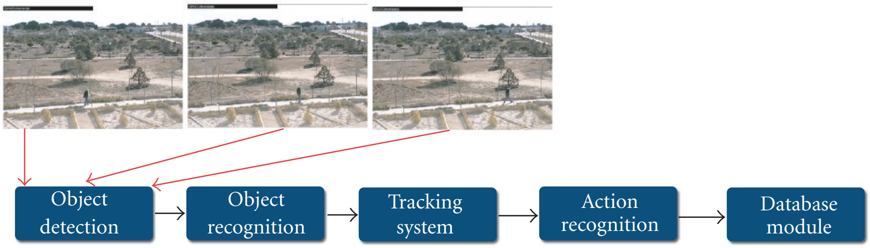


Figure 1: A generic video processing framework for an automated visual surveillance system.

developing semiautomatic tools that can help create the large set of ground truth data that is necessary to evaluate the performance of the tracking algorithms [48].

(4) Action recognition process. Since this process should recognize and understand the activities and behaviors of the tracked objects, it is a classification problem. Therefore, it involves matching a measured sequence to a precompiled library of labelled sequences that represent prototypical actions that need to be learnt by the system via training sequences. There are several approaches for matching time-varying data: dynamic time warping (DTW) [59, 60], HMM (hidden Markov models), Bayesian networks [61, 62], and declarative models [42].

(5) A database module. The final module is related to efficiently storing, indexing, and retrieving all the surveillance information gathered.

Many video surveillance systems incorporating the above techniques are currently developed and installed in real environments. Typical examples of commercial surveillance systems are DETEC [15] and Gotcha [16] or [17]. They are usually based on what is commonly called motion detectors, with the option of digital storage of the detected events (input images and time-stamped metadata). These events are usually triggered by objects appearing in the scene. Another example of a commercial system intended for outdoor applications is DETER [63] (detection of events for threat evaluation and recognition), which reports unusual movement patterns of pedestrians and vehicles in outdoor environments such as car parks. DETER consists of two parts: the computer vision module and the threat assessment module (high-level semantic recognition with off-line training and on-line threat classifier). Visual traffic surveillance for automatically identifying and describing vehicle behavior is presented in [13]. The system uses an EKF (extended Kalman filters) as a tracking module, and also includes a semantic trajectories interpretation module. For other surveillance for different applications (e.g., road traffic, ports, and railways), see [3, 6, 9–11]. A vision-based surveillance system is developed in [25] to monitor traffic flow on a road, but focusing on the detection of cyclists and pedestrians. The system consists of two main distributed processing modules: the tracking module, which processes in real time and is placed on a pole by the roadside, and the analysis module, which is performed off-line in a PC. The tracking module consists of four tasks: motion detection, filtering, feature extraction using quasi-topological features (QTC), and tracking using first-order Kalman filters. Many of these systems require a wide geographical distribution that calls for camera management and data communication. Therefore, [6] proposes combining existing surveillance traffic systems based on networks of smart cameras. The term “smart camera” (or “intelligent camera”) is normally used to refer to a camera that has processing capabilities (either in the same casing or nearby) so that event detection and event video storage can be done autonomously by the camera.

The above-mentioned techniques are necessary but not sufficient to deploy a potentially large surveillance system including networks of cameras and distributed processing capacities. Spatially distributed multisensor environments raise interesting challenges for surveillance. These challenges relate to data fusion techniques to deal with the sharing of information gathered from different types of sensors [64], communication aspects [65], security of communications [65], and sensor management. A third-generation surveillance system would provide highly automated information, as well as alarms and emergencies management. This was the stated aim of CROMATICA [8] (crowd monitoring with telematic imaging and communication assistance) followed by PRISMATICA [5] (pro-active integrated systems for security management by technological, institutional, and communication assistance). The developed system is a wide-area multisensor distributed system, receiving inputs from CCTV, local wireless camera networks, smart cards, and audio sensors. PRISMATICA then consists of a network of intelligent devices (that process sensor inputs). These devices send and receive messages to/from a central server module (called “MIPSA”). The server module coordinates device activity, archives/retrieves data and provides the interface with a human operator. Another important project is ADVISOR. It aims to assist human operators by automatically selecting, recording, and annotating images containing events of interest. ADVISOR interprets shapes and movements in scenes being viewed by the CCTV to build up a picture of the behavior of people in the scene. Although both systems are classified as distributed architectures, they have a significant key difference: PRISMATICA employs a centralized approach, whereas ADVISOR can be considered as a semi-distributed architecture. PRISMATICA is built on the concept of a main or central computer which controls and supervises the whole system. ADVISOR can be seen as a network of independent dedicated processor nodes (ADVISOR units), ruling out a single point of failure.

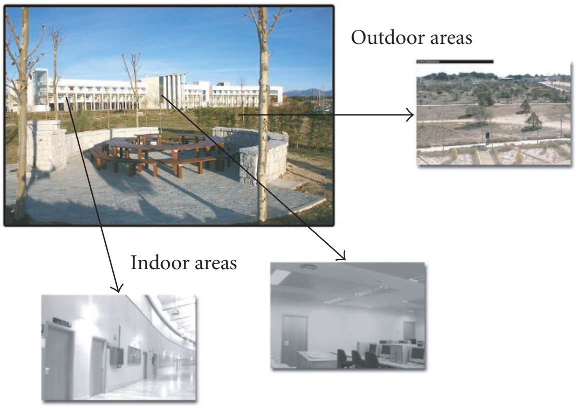


Figure 2: Several scenes captured by the cameras of our campus surveillance system. Notice that there are different areas to guard.

The design of a surveillance system with no server to avoid this centralization is reported in [66]. All the independent subsystems are completely self-contained, and all these nodes are then set up to communicate with each other without having a mutually shared communication point. As part of the VSAM project, [67] presents a multi-camera surveillance system based on the same idea as [68]: the creation of a network of "smart" sensors that are independent and autonomous vision modules. In [67], however, these sensors are able to detect and track objects, classifying the moving objects into semantic categories such as "human" or "vehicle" and identifying simple human movements such as walking. The user can interact with the system in [67].

The surveillance systems described above take advantage of progress in low-cost high-performance processors and multimedia communications. However, they do not account for the possibility of fusing information from neighboring cameras. Current research is focusing on developing surveillance systems that consist of a network of cameras (monocular, stereo, static, or PTZ (pan/tilt/zoom)) running the type of vision algorithms that we reviewed earlier, but also using information from neighboring cameras. For example, the system in [23] consists of eight cameras, eight feature server processes, and a multitracker viewer. CCN [69] (co-operative camera network) is an indoor application surveillance system that consists of a network of PTZ cameras connected to a PC and a central console to be used by a human operator. A surveillance system for a parking lot application is described in [21]. It uses static camera subsystems (SCS) and active camera subsystems (ACS). The Mahalanobis distance and Kalman filters are used for data fusion for the multitracker, as in [23]. In [68] 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. The design architecture of the system is similar to ADVISOR [7]. An interesting example of a multitracking camera surveillance system for indoor environments is presented in [57]. The system is a network of camera processing modules, each of which consists of a camera connected to a computer, and a control module, which is a PC that maintains the database of the current objects in the scene. Each camera processing module uses Kalman filters to enact the tracking process. An algorithm was developed that takes into account occlusions to divide the tracking task among the cameras by assigning the tracking to the camera that has better visibility of the object. This algorithm is implemented in the control module.

As has been illustrated, a distributed multi-camera surveillance requires knowledge about the topology of the links between the cameras that make up the system in order to recognize, understand and track an event that may be captured on one camera and to track it across other cameras. Our paper presents a framework that employs a totally deliberative process to represent the information fusion between neighboring cameras and to manage the coordination decision-making in the network.

3. MULTI-AGENT FRAMEWORK ARCHITECTURE

In this section we describe the components of our multi-agent framework architecture for designing surveillance systems. Each agent deliberatively makes decisions to carry out the system tasks coherently with other agents, considering both the information generated in its local process and the information available in the network. Transitions between areas covered by different agents will be the most important situations in this coordination process (see Figure 2).

The challenge of extracting useful data from a visual sensor network could become an immense task if it stretches to a sizeable number of cameras. Our framework operates at two logical levels. First, each camera is associated with a process that acquires current estimates and interprets its local scene. This process is partially based on a tracking system, where the detected objects are processed for recognition. A high-level representation of the interesting objects moving in the scenario is recorded to estimate their location, size, and kinematic state [70] (see Figure 3). This information is processed by different algorithms, as described in [70–72], for extraction with widely varying degrees of accuracy,

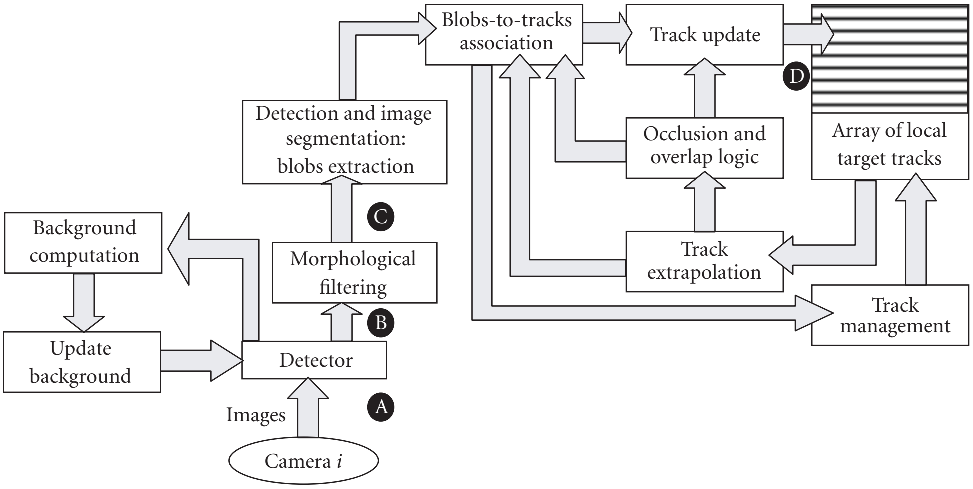


Figure 3: Structure of video surveillance system.

computational demands, and dependencies on the scene being processed. Evolutionary computation has been successfully applied to some stages of this process to fine-tune overall performance [73]. The structure of these algorithms is presented in Figure 3 and explained at length in [70]. To illustrate the process, Figure 4 shows different levels of information handled in the system stages (labelled with letters A–D in Figure 3), ranging from raw images to tracks.

Second, the information extracted must be collected and fused. The multi-camera surveillance coordination problem can be solved in a centralized way: an all-knowing central entity that makes decisions on behalf of all the cameras as is suggested in [74, 75]. However, a distributed solution may sometimes (due to scalability and fault-tolerance requirements) become an interesting alternative. Distribution is achieved through a multi-agent system, where a single software agent represents and controls each camera. Each agent only knows about some external events (partial knowledge), and has to make decisions with this limitation. Consequently, the quality of the decision cannot be optimal. Even with partial knowledge, we try to show how coordination among agents can improve the quality of decisions bringing them close to optimum.

Each camera is controlled by an agent, which will make decisions according to an internal symbolic model that represents encountered situations and mental states in the form of beliefs, desires, and intentions. As we mentioned before, our multi-agent framework takes a BDI approach [54, 76–78] to modeling camera-agents. The final goal of agents is to improve the recognition and interpretation process (object class, size, location, object kinematics) of mobile targets through cooperation, and, therefore, to improve the surveillance performance of the whole deployed camera system. The cooperation between camera-agents takes place for the purpose of improving their local information, and this is achieved by message exchange (see Figure 5). In our domain, we suggest that the beliefs, desires, and intentions of each camera-agent are the following.

(I) Beliefs

Camera-agent beliefs should represent information about the outside world, like objects that are being tracked, other known camera-agents who are geographically close and their execution state, and geographic information, including location, size and trajectory of tracked objects, location of other elements that might require special attention, such as doors and windows, and also obstacles that could occlude targets of interest (e.g., tables, closets).

(II) Desires

Camera-agents have two main desires because the final goal of a camera-agent is the correct tracking of moving objects: permanent surveillance and temporary tracking. The corresponding surveillance plan is as follows: camera-agents permanently capture images from the camera until an intruder is detected (or announced by a warning from another camera-agent). On the other hand, the tracking plan is initiated by some event (detection by camera/warning from another agent), and it runs a tracking process internally on the images from the camera until tracking is no longer possible.

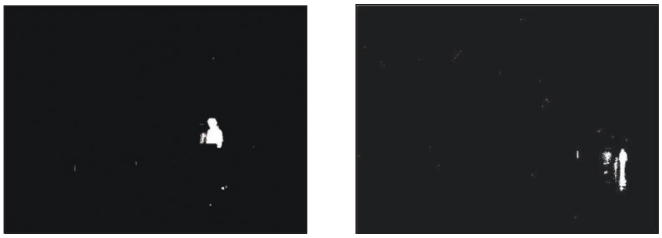
(III) Intentions

There are two basic actions: external and internal actions. External actions correspond to communication acts with other camera-agents that implement different cooperative dialogs, while internal actions involve commands to the tracking system, and even to the camera.

4. INFORMATION MANAGEMENT THROUGH CAMERA-AGENTS COORDINATION

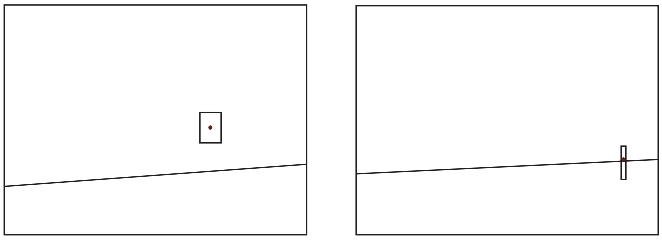
All we have discussed up to this point are the components of our framework, that is, the camera-agents. In this section we detail the problem of information management through the coordination of camera-agents. The information flowing in

(a) Original images



(b) Detected pixels

(c) Filtered images



(d) Estimated tracks

Figure 4: Information levels in the processing chain. Characters from (a) to (d) are related to the modules of **Figure 3**.

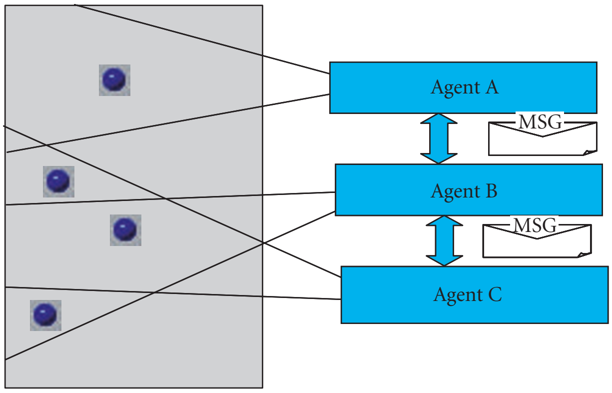


Figure 5: Overview of camera agents exchanging messages.

our multi-agent framework is used to achieve the following goals.

(1) To ensure that an object of interest is successfully tracked across the whole area to be guarded, assuring continuity and seamless transitions. Objects of interest are able to move within the restricted area and several camera-agents share part of their fields of view. When an object of interest reaches an area shared with neighboring camera-agents, they establish a dialog in order to exchange information about the object.

(2) To reason about information on objects of interest simultaneously tracked by two or more camera-agents. This kind of dialogs starts, for example, if a camera-agent loses an object of interest and queries a neighboring camera-agent about the object.

(3) To manage dependences between neighboring cameras and carry out the network tasks for use in other activities (usually surveillance tasks managed by a human operator) when the network has no objects to track.

Based on these goals, we developed the surveillance process of a generic camera-agent. As we outlined before, camera-agents may run two main types of plans: *surveillance* and *tracking*. The first plan is continuously active and governs the general surveillance of the camera's field of view. This internal process (encapsulated in another Java class, and invoked from this initial surveillance plan) consists of capturing sequential images from the camera and observing potential moving objects (intruders). When such an observation is made (an intrusion is suddenly detected), a *tracking* subplan will then be initiated for the purpose of tracking this moving object. The tracking goal is invoked taking as parameter the identification of the object. Bearing in mind that the possible goals in JADEX are perform, maintain, achieve, and query, perform seems to be the most appropriate description of its intention.

Furthermore, *tracking* plans can be fired from an internal event produced by the *surveillance* plan, but they can also be initiated by external events such as messages from other agents. This is the case of an accepted proposal of tracking from an agent that is geographically close (in the same room, or in a room linked by doors and windows with that room).

This tracking plan implementation starts an internal tracking process with the advantage of prior warning from the other agent, or with no prior knowledge about the object if it was initiated as a subgoal of the *surveillance* plan of the same agent.

Additionally, the internal process of tracking (ruled by the tracking plan) may lead to internal events on two grounds.

(a) The tracked moving object is close to a zone of limited vision (e.g., doors and windows), and the moving object is expected to move out of the camera's field of view in the near future.

(b) Or the moving object is already out of camera's field of view.

In the first case, the agent will warn the agents governing the closest cameras about the expected appearance of the moving object, starting a call-for-proposals dialog that is performed by another subgoal: "*warning about expected object dialog.*"

In the second case, the agent queries other agents that could possibly view the moving object that disappeared to determine whether or not the moving object really did leave the camera's field of view (and, therefore, whether or not the internal tracking process should be terminated). The implementation of the query dialog is performed by another subgoal: "*looking for lost object dialog.*"

Camera agents also require another plan to confirm/disconfirm the presence of a given moving object when another agent submits a query about the object. This plan just evaluates whether or not the moving object is visible from the camera, and then reports the result of the evaluation to the other agent.

Finally, external (human) intervention would cause a querying plan to be fired (asking for permission to be temporarily unavailable: "*requesting for a break dialog*"), in a surveillance, as many warning plans would be fired as objects were currently being tracked by the agent.

In conclusion, the hierarchy of surveillance domain plans is illustrated in Figure 6.

Since these messages comply with the FIPA standard, they include a performative to represent the intention of the respective communicative act. These performatives can be: accept, agree, cancel, propose, confirm, disconfirm, failure, inform, propagate, propose, query-if, refuse, reject proposal, request, call for proposals, and so forth.

Broadly speaking, three main dialogs can take place between agents.

(i) "*Warning about expected object dialog.*" It intends to warn the receiving agent about the expected future presence of a moving object. The goal is that the receiving agent initializes a tracking plan for this moving object. This warning takes the form of a proposal.

(ii) "*Looking for lost object dialog.*" It anticipates a confirmation of the presence of a moving object in the receiving agent's field of view. It would usually complement the first dialog, but it can be produced standalone.

(iii) "*Requesting for a break dialog.*" In this dialog the sending agent asks the receiving agent for permission to become temporarily unavailable, and objects placed in shared areas should be tracked by the receiving agent. This dialog may also include the "*warning about expected object dialog,*" since the receiving agent may want to warn the sending agent about its tracked objects that are likely to be in the field of view of the sending agent according to its current trajectory. Finally, the receiving agent will confirm/retract its temporary unavailability.

Next, we detail some aspects about these dialogs.

4.1. Warning about expected object dialog

The first dialog would take place if agents expect some circumstances in the very near future that would prevent the object from being tracked. These circumstances occur when the moving target is close to zones that cannot be tracked because they are out of the field of view of the camera controlled by the agent in question.

Since several receiving agents are often possible trackers of the moving object, the sending agent (who is currently tracking the movement of the object) sends a "call for proposals" to all of the candidates. The FIPA "call for proposals" message contains an action expression denoting the action "act" to be done, and a referential expression defining a proposition that gives the preconditions (in the form of a single-parameter function f(x)) on the action "act." In other words, the sending agent asks the receiving agent: "will you perform action "act" on object "x" when "f(x)" holds?" Where "x" stands for the "moving object," "act" stands for

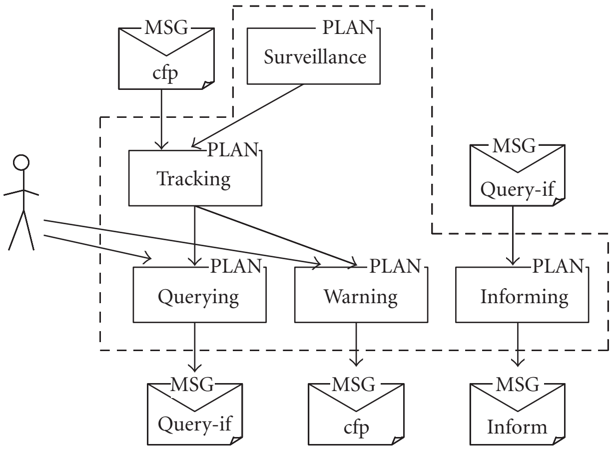


Figure 6: Relationship between received messages and fired plans.

“tracking,” and “f(x)” should be determined by the receiving agent. In normal usage, the agent responding to a cfp should answer with a proposition giving the value of the precondition expression. An example of this message would be

```
(cfp
:sender (agent ?j)
:receiver (agent ?i)
:content (track (object ?x))
:reply-with cfpx
)
```

Where variables ?i, ?j, and ?x correspond to JAVA objects, whose inclusion and extraction from FIPA messages are facilitated by JADEX. In our case of surveillance, objects would allow them to be correctly identified for sending and receiving agents, for instance, using global positioning, or references to shared visual elements such as doors and windows that link one room with another.

After the reception of a cfp message, one of the receiving agents would volunteer as the tracker of the given moving object. So the next FIPA performative should be “propose” where the proposer (the sender of the proposal) informs the receiver that the proposer will adopt the intention to perform the action once the given precondition is met. Preconditions can be: the door is finally opened, the object is finally viewed by the camera, and so forth. The expression of all such possible preconditions should be previously defined and shared by all agents in an ontology. An example of this message would be

```
(propose
:sender (agent ?i)
:receiver (agent ?j)
:content (track (object ?x )) (visible (object ?x))
:ontology surveillance
:reply-with proposex
:in-reply-to cfpx
)
```

Then, the receiver of the proposal (who initially sent the cfp) should accept the proposal with the corresponding FIPA performative. Accept-proposal is a general-purpose acceptance of a proposal that has previously been submitted (typically through a propose act). The agent sending the acceptance informs the receiver that it intends the receiving agent to perform the action (at some point in the future), once the given precondition is, or becomes, true.

```
(accept-proposal
:sender (agent ?i)
:receiver (agent ?j))
:content (track (object ?x )) (visible (object ?x))
:ontology surveillance
:in-reply-to proposex
)
```

With the acceptance of the proposal the *warning dialog* between agents ends.

4.2. *Looking for lost object dialog*

The second dialog would often take place when some unexpected circumstances suddenly occur: the moving agent disappears from a camera-agent’s field of view, but this was not predicted/observed (e.g., the moving agent may be hidden behind a closet or table). This dialog is intended to get a confirmation that another agent is viewing the moving object. Therefore, the first message is a query to a camera-agent that is the potential viewer of the moving object. The corresponding FIPA performative is “query-if,” that is, the act of asking another agent whether (it believes that) a given proposition is true. The sending agent is requesting the receiver to tell it whether the proposition is true. In our case the proposition is that the moving object is visible for the receiving agent. The agent performing the query-if act has no knowledge of the truth value of the proposition, and believes that the other agent can inform the querying agent about it. So the receiving agent would answer with an “inform” FIPA communicative act:

```
(query-if
:sender (agent ?j)
:receiver (agent ?i)
:content (visible (object ?x))
:reply-with queryx
)
(inform
:sender (agent ?i)
:receiver (agent ?j))
:content (not (visible (object ?x)))
:in-reply-to queryx
)
```

4.3. *Requesting for a break dialog*

The third dialog would take place when an agent needs to leave the automated surveillance plan, perhaps to let humans control the camera manually, for instance, to focus on some details (zoom). Therefore, all objects being tracked would be lost for a while.

This dialog intends to let other agents know about its temporary unavailability, asking about the convenience of such unavailability. The corresponding FIPA performative is "query-if," that is, the act of asking another agent whether (it believes that) a given proposition is true. The sending agent is requesting the receiver to inform it of the truth of the proposition. In our case, the proposition is that there is no object coming towards the field of view of the sending agent in the very near future. The agent performing the query-if act has no knowledge of the truth value of the proposition, and believes that the other agent can inform the querying agent about it. So the receiving agent would answer with an "inform" FIPA communicative act:

```
(query-if
:sender (agent ?j)
:receiver (agent ?i)
:content (is-anyone-coming?)
:reply-with queryanyone
)
(inform
:sender (agent ?i)
:receiver (agent ?j))
:content ((object ?x)))
:in-reply-to queryanyone
)
```

Also objects placed in shared areas should be then tracked by the receiving agent. Consequently, for each object located in such a shared area that is currently being tracked by the sending agent, a cfp dialog (the first type) would take place to leave the tracking of that object to the receiving agent.

Therefore, these seven messages are the main stream of communication acts in our surveillance domain. There are also others, such as the rejection of proposals from agents in reply to cfp messages because another agent already submitted a proposal, other auxiliary messages due to delays, misunderstandings, and so forth, but they are not detailed here for brevity, although they also comply with the FIPA standard.

5. APPLICATION SCENARIOS OF THE MULTI-AGENT FRAMEWORK

In order to illustrate the capability of our multi-agent framework and evaluate its performance on coordination tasks, we have applied it to two practical scenarios and compared the results against a surveillance system without coordination mechanisms.

Based on the agent framework described above, we particularized the beliefs for creating new scenarios. In the following, we briefly present the functionality and tailoring for the two scenarios.

(1) The first application is an indoor application in which two agent-cameras detect intruders in a restricted room. The first agent controls the corridor leading to the room. Once it has detected an intruder and checked that it is close to the door to the room, the corridor agent sends a message to alert the agent-camera inside the room. The message contains not only the warning that there is an intruder, but also the information about this intruder: size, kinematics, and so forth. This is very useful for the room agent because the restricted room has many objects that may occlude the stranger and the lights might deform the person and confuse the agent. Therefore, the main dialog between agents uses the "*warning about expected object dialog*" and "*looking for lost object dialog.*" With this scenario, we demonstrate that our multi-agent framework is more reliable and robust than the one without agent coordination.

(2) The second scenario is an outdoor application in which two agent-cameras control pedestrians (considered also as intruders) walking down a footpath. Both agents share an overlapped area in their field of view. In this particular scenario, the pedestrians walk from left to right, so the left agent warns to the right agent about the presence of an intruder when it reaches the shared area. This conversation is carried out by a "*warning about expected object dialog.*" Occasionally, if there are no messages from the left agent reporting new intruders, the right agent can ask the left agent for temporary disconnection from the surveillance system to do another activity using the "*requesting for a break dialog.*" Thanks to the coordination between the two agents, we illustrate that our framework is capable of multitasking without affecting the global surveillance activity.

Finally, we present a set of evaluation metrics to compute the performance and assess the advantages and disadvantages of using a multi-agent framework as compared with architectures without agent coordination.

5.1. Indoor scenario

In the first scenario, the system must be able to detect and track an intruder using cameras covering a room and an access corridor (see Figure 7). This is basically a case of detecting and tracking intruders in a restricted indoor area, where the system must reliably detect the presence of intruders and guarantee continued tracking of their movement around the building. Furthermore, the communication between agents should contribute to providing a more reliable and robust surveillance system. In order to show this improvement we will evaluate a set of video samples to get statistically significant results. In this particular case, a corridor agent passes all the available information about the intruder on to a room agent. Thus, the room agent reconstructs the real track that is usually corrupted by the occlusions and shadows present in the room. One characteristic of distributed indoor surveillance, compared with open environments, is the presence of multiple transitions between areas exclusively covered by different cameras, such as corridors and rooms, with very quick handovers.

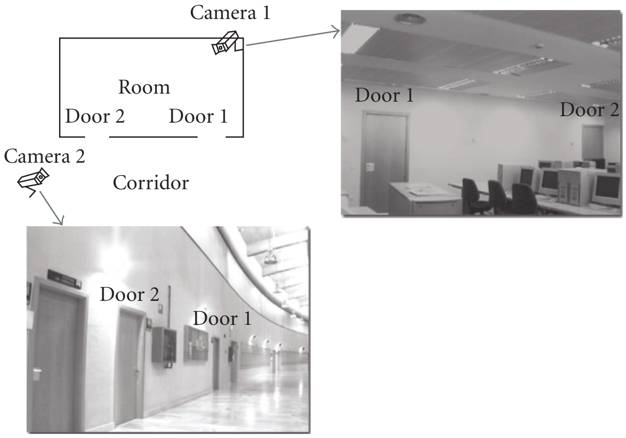


Figure 7: Indoor scenario. There are two camera-agents; one (camera 1) is guarding a room with two doors and the other (camera 2) is placed outside the room, in a corridor.

5.1.1. *BDI representation*

The known context for this scenario containing two BDI agents is based on the following premises.

1. There is a single intruder. The system would work with more than one intruder, but we simplify this condition to make the evaluation easier.
2. The intruder moves from the corridor to the room through either of the doors leading on the same room.
3. One camera can observe the whole room and the other one the corridor.

Based on these assumptions, we defined the following beliefs, which particularize the BDI framework to this specific scenario.

1. The agents are close to each other and to the doors that link our room with the areas they cover (corridors) through the tuple (agent id, list of door ids). They are consulted to determine who is to receive the cfp message when the moving object is close to any door and to answer the query-if message with the corresponding inform message.
2. Location of the moving objects with three possible values: not-visible, close-to-door (door-id), and visible. The close-to-door value in this belief fires the execution of the warning plan (cfp message).
3. Description of the moving objects (coordinates of center of gravity and size) that are received from the cfp message and that are input to the internal tracking process to improve initial predictions.
4. Description of the doors (4 coordinates of its squares), which are input to the internal tracking process to improve initial predictions.

These beliefs are enough to run an execution where a camera-agent (identified as "corridor") is located in a corridor that is tracking the movement of an intruder (identified as "intruder"), and another camera-agent (identified as "lab") is located in a lab linked to the corridor through two doors (identified as "door0" and "door1").

Therefore, the corridor agent is executing both main plans: tracking and surveillance, and it also has these initial beliefs: "close-agent (lab, {door0, door1})" and "location-intruder (intruder, visible)."

And the room agent is executing just the surveillance plan, and it also has these initial beliefs: "close-agent (corridor, {door0, door1})" and "location-intruder (intruder, not-visible)."

When the intruder moves close to the door identified as door1, then the internal tracking process points out that the belief location of the intruder changes its value to "location-intruder (intruder, close-to-door(door1))." This change initiates a warning plan (starts the "*warning about expected object dialog*"), which sends a cfp message to the lab agent:

```
(cfp
:sender (corridor)
:receiver (lab)
:content (track (intruder-at (intruder, door1)))
:reply-with cfpx
)
```

Then, the room agent starts a tracking plan, because it now expects the intruder to enter through door1. When this intruder enters the room, the tracking process points out a change in the belief of the intruder's location. It changes from "not-visible" to "visible." This change allows the right response to the query-if message that the corridor agent will send to execute the querying plan activated when this agent loses sight of the intruder. As soon as the query-if message is received from the corridor agent, the room agent executes the informing plan in response to that query ("*looking for lost object dialog*"). The dynamic schema of the "warning about expected object dialog" and "looking for lost object dialog" is depicted in Figure 8.

5.1.2. Experimental evaluations

Now, we are going to evaluate whether there is any improvement in the surveillance system through agent coordination as compared with the isolated operation of a particular node. An agent surveillance plan is able to follow all kinds of targets and their different movements across the whole camera plane. The effect of using flow information coming from neighbor agents should increase the reliability of agent estimations, as it will be assessed throughout this section.

For the purpose of evaluating the tracking system, let us suppose that the intruder enters the room and moves along the wall from door2 to door1. This trajectory is used as *ground truth* exclusively to assess system performance under these conditions (it is not information available in the agent). We have selected 15 recorded situations of this intrusion action, which we have evaluated with and without information exchange between both agents. The quality measures of both experiments were computed averaging tracking results of 15 video sequences and the path followed by the intruder.

We have previously applied evaluation metrics to assess video surveillance systems [72]. In our evaluation system, each time a track is initiated or updated by the agent tracking plan, the results are stored for analysis by the evaluation system. To get a more detailed idea of system performance, the agent-camera plane is divided into 10 zones (see Figure 9). Each zone is defined as a fixed number of pixels on the x-axis, 10% of the horizontal size of the image. The horizontal component has been selected to analyze the metrics because it is the main coordinate along which the objects move in this particular study.

The metrics that we have applied to both experiments are the following.

(a) Initialization: this is the number of frame in which the intruder is detected by the agent tracking plan.

(b) Absolute area error: this is computed by calculating the area of the detected track. It is important to measure the absolute area to get an idea of what the camera is really tracking. For example, in this case, the lights of the room make the intruder look bigger than her real size due to the projected shadow. Therefore, the uncoordinated cameras track not only the shape of the person but also her shadow. The coordination messages overcome this problem by adapting the track to the real size.

(c) Transversal error ($d(P, r)$): it is defined as the distance between the center of the bounding rectangle (P) and the segment (r), which is considered as ground truth (see Figure 10).

(d) Interframe area variation: this metric is defined as the variation of area between the current update and the previous update of the track under study. It is required to check that the previous track exists. Otherwise, the value of this metric is zero.

(e) Continuity faults: the continuity faults metric is only measured inside a gate defined by the user. This gate is chosen so as to represent the area in which no new tracks can appear or disappear, because the intruder has already turned up on the right side of the image. This metric checks whether or not a current track inside the gate existed before. If the track did not exist, it means that this track was lost by the agent tracking plan and recovered in a subsequent frame. This behavior must be computed as a continuity fault. This continuity metric is a counter, where one unit is added every time a continuity fault occurs.

(f) Number of tracked objects: it is known that there is only one intruder per video, but the agent tracking plan may fail and sometimes follow more than one or zero. Thus, every time a track is initiated, the agent surveillance plan marks it with a unique identifier. This metric consists of a counter, which is increased by one unit every time a new object with a new identifier appears in the area under study. After the evaluation of all the videos, this metric is normalized by the total number.

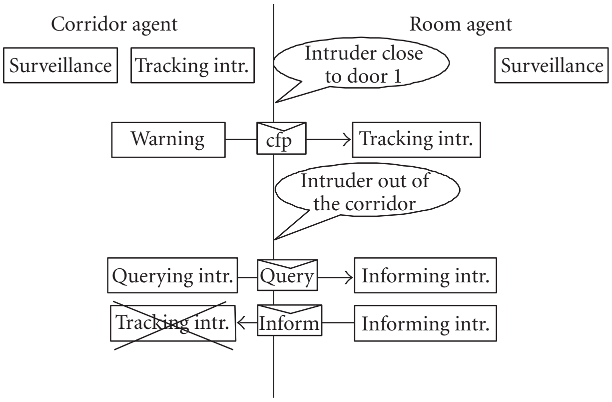


FIGURE 8

FIGURE 9: Segmentation of each frame into ten zones for better measurement accuracy.

5.1.3. Performance results

The following tables and graphs compare tracking system performance with and without the agent coordination operating in the system.

First of all, we find from Table 1 that the system inside the room initializes the intruder track as soon as a message with information about the intruder is available. Some frames later, the initialization is confirmed when the person enters the room. Otherwise, the initialized track must

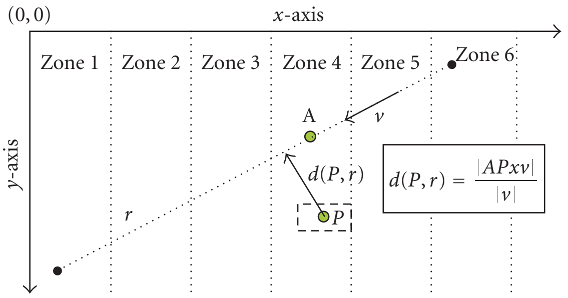


FIGURE 10: Distance from a target to a reference path.

TABLE 1: Comparison of the initialization of an intruder track for the two available systems. The system with agent architecture initializes the track when a message from the outside camera is received by the inside camera (frame number 1).

Recorded video number	Initialization frame	
	System without agent architecture	System with agent architecture
1	20	1
2	19	1
3	19	1
4	22	1
5	40	1
6	23	1
7	18	1
8	22	1
9	24	1
10	18	1
11	17	1
12	24	1
13	18	1
14	26	1
15	15	1

be removed. On the other hand, if the tracking system has no previous knowledge, the initialization will be carried out after the agent-camera surveillance plan detects the intruder.

Second, the absolute area error of the tracked object with activated agent coordination is almost constant as is clear from Figure 11(b), compared with the isolated case (a). We find that the area on Figure 11(b) has a much lower variation and is almost constant compared with the situation on Figure 11(a). The graphs in Figures 11 and 14 have a solid line indicating the mean value, two dashed lines around the solid line representing standard deviation ($+-1\sigma$), and two dotted lines specifying the maximum and minimum values. The graphs are divided horizontally into 10 zones representing the whole area covered by the agent surveillance plan.

The effect on the estimated area is because the corridor agent-camera sends stable information about the location and size of the intruder to the agent-camera in the room. This agent quickly initializes and rebuilds the representation, which is updated later from the observation generated by the actual camera. Thus, the surveillance system processes some blobs that are added to with the knowledge passed in the message: height and width of the person. Therefore, the surveillance system tracks the available blobs (some of which are impossible to detect due to occlusions) and reconstructs the original size. Furthermore, this computation stops shadows and reflections from being taken into account, because this spurious information tracked by the surveillance system will not fit in with the previous information and will be discarded. Figure 12 shows the points marked as pixels in motion. Many of these points are spurious information due to the light coming into the room when the door is opened and the reflection of this light on the wall. Furthermore, the intruder is partially occluded by the tables and computers. The system is able to reconstruct the position and the size of the intruder and remove the incorrect information.

Obviously, the interframe area variation, or the variation of the area from the last to the current update of the track under study, of our new system is very low, since the room agent has information about the location and size of the intruder, and this is used for its estimations.

The following pictures give us a clear idea of system performance. Figures 13(a) and 13(b) are two frames of a video sequence, Figures 13(c) and 13(d) show the points marked as pixels in motion, and Figures 13(e) and 13(f) contain the system output. Thus, Figure 13(c) shows the blobs processed by the system for Figure 13(a). The system cannot capture any more blobs of the intruder, as there are obstacles (tables and computers) in the way. The surveillance system outputs the intruder track that is depicted in Figure 13(e) by the smaller rectangle. Nevertheless, the coordinated agent rebuilds the intruder track using the previous knowledge of the intruder's size. The same process is shown for Figure 13(b). In this case, the obstacles allow the surveillance system to capture more pixels so that the system has to rebuild fewer parts of the intruder.

The transversal error with respect to ground truth is depicted for both cases in Figure 14. It is clear that the error is almost zero for the second architecture (Figure 14(b)) because the track is adjusted using the previous knowledge. As we said before, the system takes the track output by the surveillance system and rebuilds it using the intruder's characteristics. In both cases, the system considers the line defined by the centers of mass of the whole person as ground truth, that is, the centers of the reconstructed tracks from door 2 to door number 1.

In Figure 15, the metric shows that our new system is more robust as there are no continuity faults. On the other hand, the system based on the surveillance system only does have some continuity faults due to a poor initialization with occluded images when the intruder enters the room.

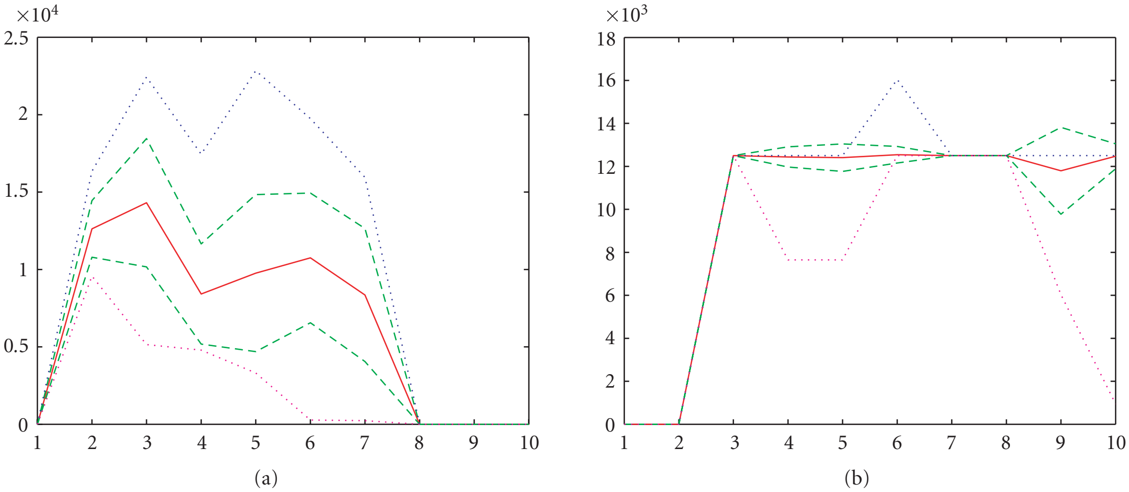


Figure 11: Absolute area error for the architecture without (a) and with (b) agent coordination.

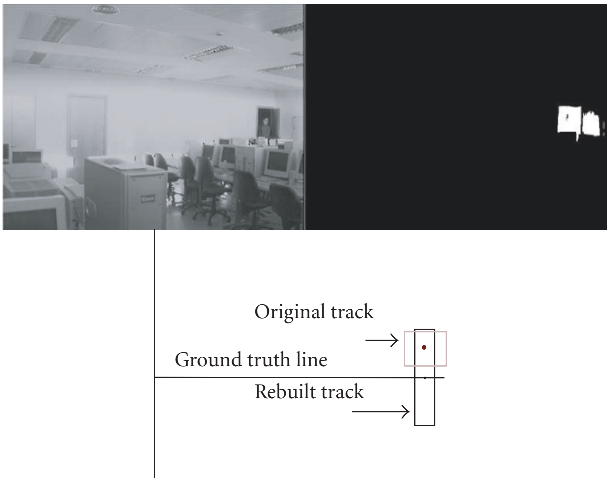


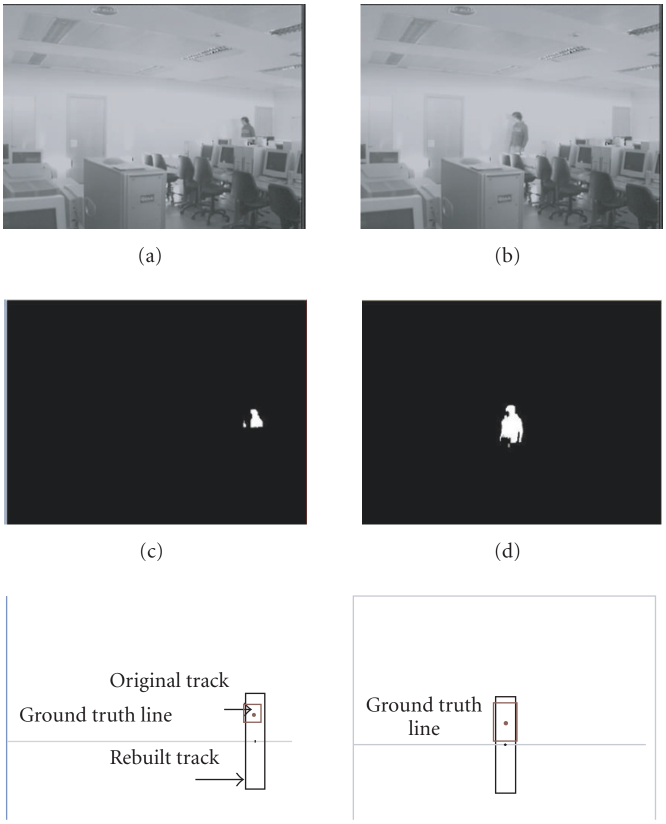
Figure 12: Reconstruction of the track based on previous knowledge.

Finally, in Figure 16, the number of tracked objects shows that the system with agent coordination stores a correct representation (one intruder) in zones 8, 9, and 10, which are the areas close to door number 2, and makes a smooth transition to actual detections (from zone 7 to the left). This is because the system initializes the intruder track from the very beginning, while this initialization is delayed considerably in the system without agent coordination.

5.2. Outdoor scenario

We now describe the second scenario in which coordinated surveillance has been applied. There are two cameras aimed at a footpath and their goal is to detect and track pedestrians (they could also be considered intruders). Both cameras share an area as depicted in Figure 17. The moment the pedestrian reaches the shared area, the right agent (camera 1) starts a "*warning about expected object dialog*" with the left agent (camera 2).

Nevertheless, the left agent can carry out other actions such as manual operation by a human user, implying that it stops the process of tracking pedestrians on this side of footpath. To avoid a disruption in the surveillance service provided by the two cameras, the left agent asks the right agent if there are any pedestrians in its field of view beforehand. That is generally done by means of a manual operator and using a "*requesting for a break dialog.*" The right agent replies to the left agent, sending a message in which it specifies whether the left agent is allowed to do another action. Therefore, whereas the main advantage of using agent coordination in scenario 1 is to improve system

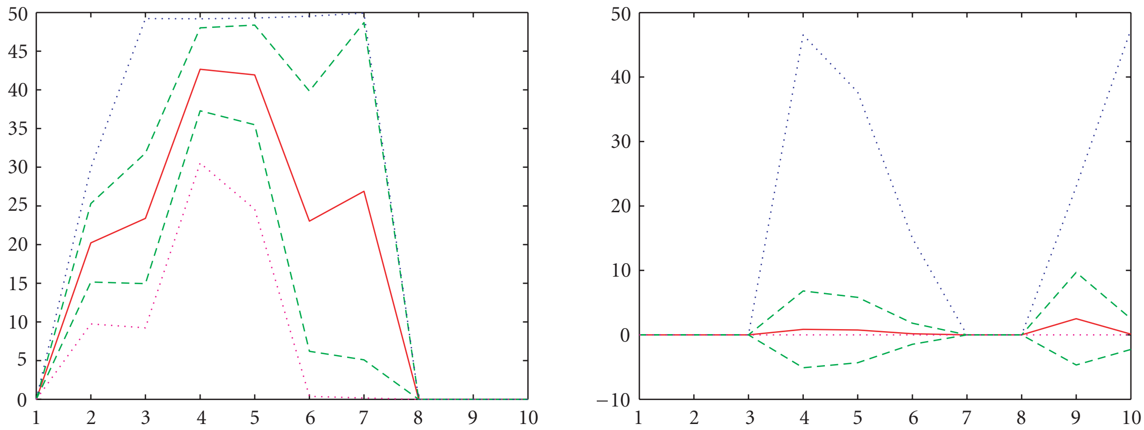


(a) (b)

(c) (d)

(e) (f)

Figure 13: System performance.



(a) (b)

Figure 14: Transversal error for the architecture without (a) and with (b) agent coordination.

performance, the main advantage illustrated in this scenario is the possibility of extending the left agent's functionality. In other words, by means of connection and disconnection actions, the left agent can carry out the main task of surveillance and other activities, such as zoom or scanning of other areas. Obviously, this agent-governed setup of the visual network, in which the interaction of cameras with human operator takes a lower priority than the performance of automatic surveillance tasks, can be switched to fully manual operation when the human urgently needs to have control of the cameras (e.g., in an emergency).

In this particular case, the surveillance system is deployed outdoors and we had to adapt the system in order to stop

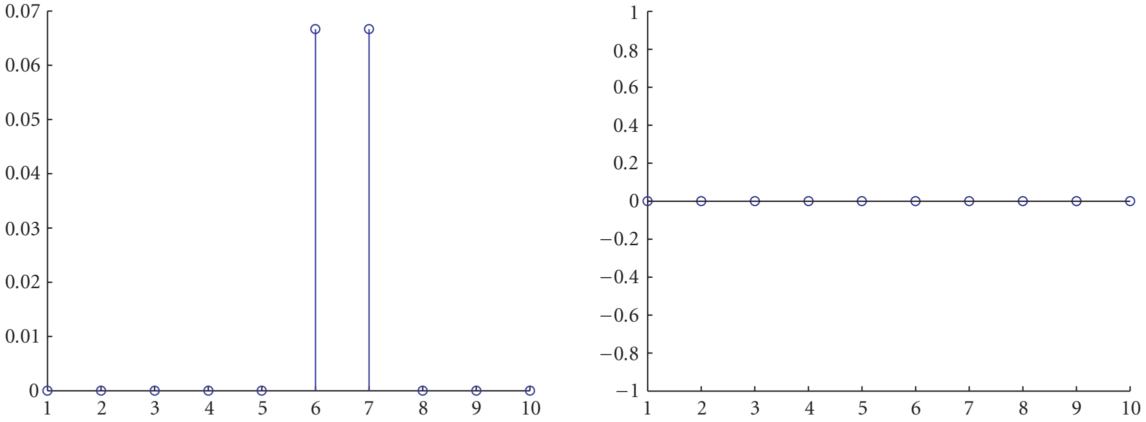


FIGURE 15: Continuity faults for both architectures.

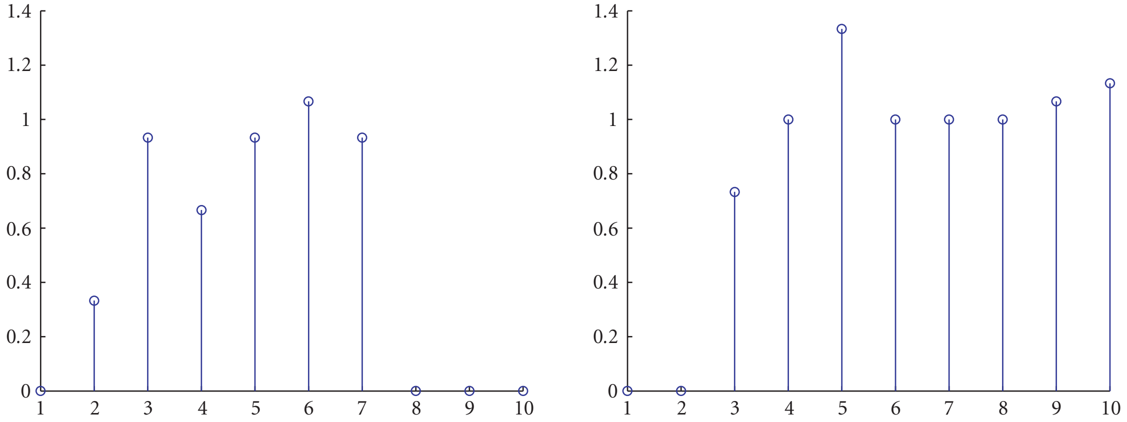


FIGURE 16: Number of tracked objects.

some incorrect detection due to the movement of trees and plants and noise.

5.2.1. BDI representation

In this scenario, we also had to make several simplifications of the general problem. For instance, we assume the following.

(1) There are only two cameras with a shared area.
(2) There are three moving objects, one of them in a shared area, and one object for the exclusive field of view of each agent.
(3) One of the objects is moving from the field of view of one agent to the other.

Based on these assumptions and using our agent coordination framework, we particularize the beliefs for this scenario.

(1) The agents are close to each others.
(2) The shared area that links the field of view of one agent with the field of view of the respective agent.
(3) Location of the moving object with three possible values: not-visible, shared area identifier, and exclusive-zone.
(4) Description of the moving object (coordinates of the center of gravity and trajectory).

These beliefs are enough to run an execution where there is a camera-agent (identified as “left”) located on the left side of the scenario that is tracking the movement of two objects (identified as “intruder0” and “intruder1”), and there is another camera-agent (identified as “right”) located on the right side of the scenario (identified as “right”) that is tracking the movement of one object (identified as “intruder2”) and there is an overlap with some of the field of view of the left agent (identified as “overlap0”).

Therefore, the left agent is executing the following plans: two tracking plans for pedestrian0 and pedestrian1, respectively, together with a surveillance plan. It also has these initial beliefs: “close-agent (right, overlap0)” and “location-pedestrian (pedestrian0, exclusive-zone) (pedestrian1, overlap0).”

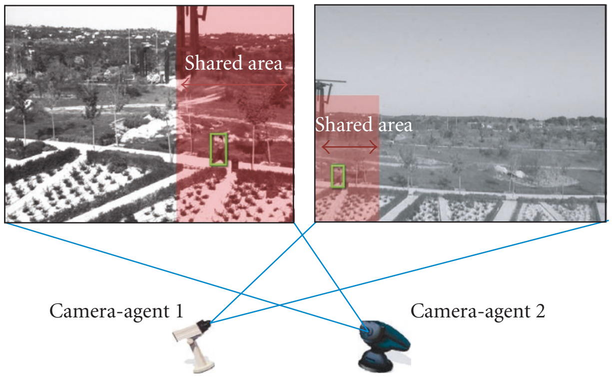


FIGURE 17: Layout for scenario 2. There are two camera-agents sharing an overlapping zone labelled as "shared area."

And the right agent is executing just one tracking plan for pedestrian2 and a surveillance plan. It has also these initial beliefs: "close-agent (left, overlap0)" and "location-pedestrian (pedestrian2, exclusive-zone)."

When the left agent receives an external event (possibly caused by a human operator) requesting manual control of the left camera, it sends a query-if message to the right agent asking for permission and also a cfp message for object pedestrian1 to be tracked in advance by the right agent since it is located in the shared zone "overlap0."

```
(query-if
:sender (agent left)
:receiver (agent right)
:content (is-anyone-coming?)
:reply-with queryanyone
)
(cfp
:sender (agent left)
:receiver (agent right)
:content (track (pedestrian-at (pedestrian1, overlap0)))
:reply-with cfpx
)
```

On the other hand, the right agent will answer the cfp message with the respective propose message to be accepted by the left agent with an accept-proposal message. Furthermore, the query-if message will be answered by an inform message, letting the left agent know about pedestrian2 since this object is moving towards the left agent.

```
(inform
:sender (agent right)
:receiver (agent left))
:content ((object pedestrian2))(mseg-expected 30)
:in-reply-to queryanyone
:reply-with informcoming
)
```

Finally, the left agent will make a decision (confirm/disconfirm) on its temporary unavailability. For instance, a confirm message including the information received about the pedestrian that is moving towards it in the content attribute of the message. The dynamic schema of the "requesting for a break dialog " is depicted in Figure 18.

```
(confirm
:sender (agent left)
:receiver (agent right))
:content ((object pedestrian2))(mseg-expected 30)
:in-reply-to informcoming
)
```

5.2.2. *Experimental evaluations*

For evaluation purposes, we consider that the pedestrians appear on the right side of the scene, and move from right to left. As mentioned, both cameras have a common area in their field of view, which is called "shared area." This common area allows the two cameras to track the targets simultaneously. This turns out to be very useful when the second camera is carrying out other task (i.e., focus on the face of another previous target to try to identify him/her), and it needs some extra time to go back to track the new pedestrian that camera 1 has indicated.

Once the right agent has detected a pedestrian, it calculates its size, location, and velocity. Based on these data, the right agent computes the seconds that it will take the pedestrian reach the shared area. This operation is very simple: a subtraction of the current pixel from the one in which the common area starts, divided by the velocity in pixels per second, where both the position (pixels) and the velocity (pixels per second) are estimated by the Kalman filter. Thus, this is the time that the left agent has to perform the other task before going back to its original position in order to track the pedestrian indicated by the right agent.

For the experiments, we recorded 14 videos. The pedestrian has a very similar velocity in eight videos, whereas, velocity increases from one scenario to the next one in the others. The mean velocity in each video is shown in Table 2.

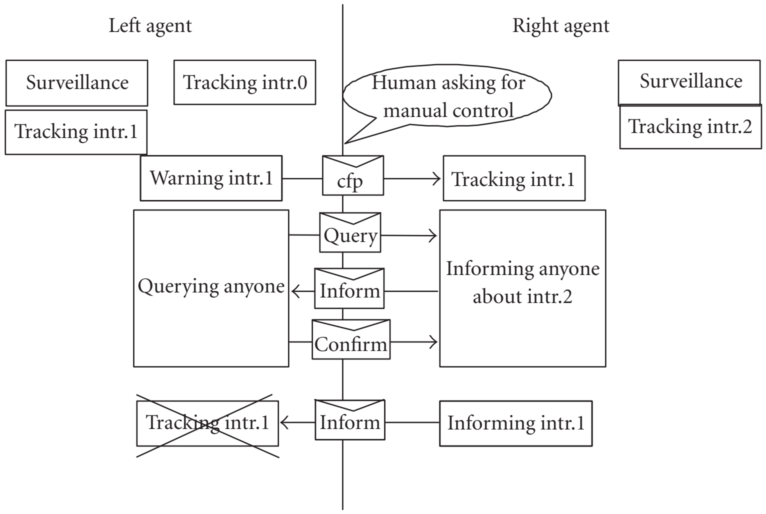


Figure 18

Table 2: Mean velocity of pedestrians in videos.

Target ID	Mean velocity (pixels/s)
1	−47.74535809
2	−43.02059497
3	−45.10638298
4	−49.1646778
5	−50.57208238
6	−47.84482759
7	−47.74590164
8	−49.6
9	−55.09138381
10	−72.04301075
11	−74.75728155
12	−128
13	−181.25
14	−215.2173913

Therefore, the faster the pedestrian moves, the less time the left agent has to carry out the other task, as is shown in Figure 19, which has been computed using the above formula.

To check the effect of the coordination between the two cameras, Figure 21 shows what would happen if the information about the new pedestrian tracked by the right agent is not shared with the left agent.

To do this, we divided the left agent's image into 10 equal zones, as we did in the evaluation of the previous scenario (Figure 20).

The experiment is composed of the following steps. First, the left agent is going to do another task and it will stop the surveillance activity without asking the right camera about nearby pedestrians. Therefore, the left camera will lose the field of view shown above to do a parallel action, that is, zoom in on a distant object. Then, while the left agent is carrying out the other activity, a pedestrian approaches the shared area. The left agent is not aware of the approaching pedestrian, and goes on with its parallel task, as we have supposed there is no coordination between the two cameras.

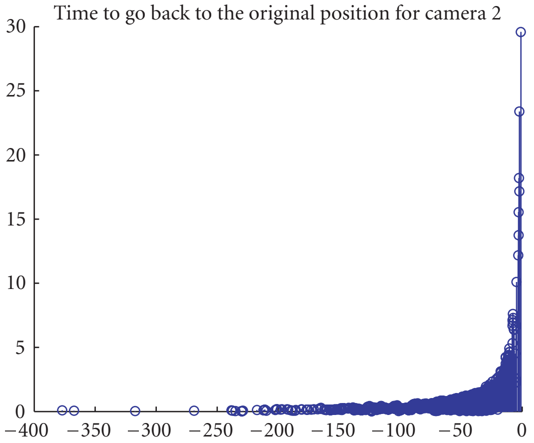


Figure 19: Seconds remaining for the left agent as a function of the velocity (pixels per second) of the target detected by the right agent.

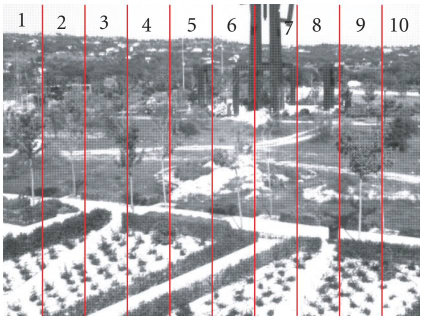


Figure 20

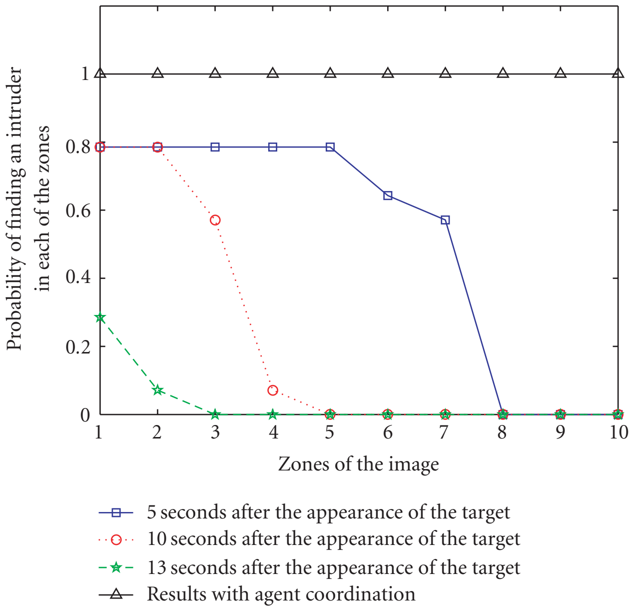


Figure 21: Probability of detecting a pedestrian in any of the zones into which the digital image is divided.

Then, the pedestrian comes into the field of view where it should be covered by the left camera and is therefore not detected. The following graph shows the probability of detecting a track in each of the zones within the field of view, supposing the left camera returns to the surveillance position 5 (line with squares), 10 (line with circles), or 13 seconds (line with stars) after the pedestrian appeared in the scene. This probability depends of the mean velocity of each pedestrian. Moreover, the line marked with triangles shows the probability of detecting a track when the agent coordination is used. We can check that no target is lost, whereas the maximum of probability of detection without coordination is 0.8. That means that from the 14 targets, at least two of them are fast enough for not being detected.

If we had used coordination between agents, the left agent would have asked for permission to carry out another activity and disconnect surveillance. The right agent would have replied, reporting the time remaining for the pedestrian to appear. Therefore, the left agent would have returned to the surveillance position in time to track the pedestrian. Then, as we said before, the graph in Figure 21 is the straight line with probability 1 in all the zones.

6. CONCLUSIONS

In this paper a multi-agent framework has been applied to the management of a surveillance system using a visual sensor network. We have described how the use of software agents allows more robust and decentralized system to be designed, where management is distributed between the different camera-agents. The architecture of each agent and its level of reasoning have been presented, as well as the mechanism (agent dialogs) implemented for coordination. Coordination enhances the continuous tracking of objects of interest within the covered areas, improves the knowledge inferred from information captured at different nodes, and extends surveillance functionalities through an effective management of network interdependences to carry out the tasks.

These improvements have been shown with the framework operating in a surveillance space (indoor and outdoor configuration for a university campus) using several numeric performance metrics. The software agents' ability to represent real situations has been analyzed, as well as how the exchanged information improves the coordination between the camera agents, thereby enhancing the overall performance and functionalities of the network.

ACKNOWLEDGMENTS

This work was funded by projects CICYT TSI2005-07344, CICYT TEC2005-07186, and CAM MADRINET S-0505/TIC/0255.

REFERENCES

[1] Airport Surface Detection Equipment Model X (ASDE-X), http://www.sensis.com/docs/128.

[2] M. E. Weber and M. L. Stone, "Low altitude wind shear detection using airport surveillance radars," *IEEE Aerospace and Electronic Systems Magazine*, vol. 10, no. 6, pp. 3–9, 1995.

[3] A. Pozzobon, G. Sciutto, and V. Recagno, "Security in ports: the user requirements for surveillance system," in *Advanced Video-Based Surveillance Systems*, C. S. Regazzoni, G. Fabri, and G. Vernazza, Eds., Kluwer Academic, Boston, Mass, USA, 1998.

[4] P. Avis, "Surveillance and Canadian maritime domestic security," *Canadian Military Journal*, vol. 4, no. 1, pp. 9–15, 2003.

[5] B. P. L. Lo, J. Sun, and S. A. Velastin, "Fusing visual and audio information in a distributed intelligent surveillance system for public transport systems," *Acta Automatica Sinica*, vol. 29, no. 3, pp. 393–407, 2003.

[6] C. Nwagboso, "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, Kluwer Academic, Boston, Mass, USA, 1998.

[7] ADVISOR specification documents (internal classification 2001).

[8] http://dilnxsrv.king.ac.uk/cromatica/.

[9] N. Ronetti and C. Dambra, "Railway station surveillance: the Italian case," in *Multimedia Video Based Surveillance Systems*, G. L. Foresti, P. Mahonen, and C. S. Regazzoni, Eds., pp. 13–20, Kluwer Academic, Boston, Mass, USA, 2000.

[10] M. Pellegrini and P. Tonani, "Highway traffic monitoring," in *Advanced Video-Based Surveillance Systems*, C. S. Regazzoni, G. Fabri, and G. Vernazza, Eds., Kluwer Academic, Boston, Mass, USA, 1998.

[11] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 495–502, San Juan, Puerto Rico, USA, June 1997.

[12] Z. Zhi-Hong, "Lane detection and car tracking on the highway," *Acta Automatica Sinica*, vol. 29, no. 3, pp. 450–456, 2003.

[13] L. Jian-Guang, L. Qi-Feing, T. Tie-Niu, and H. Wei-Ming, "3-D model based visual traffic surveillance," *Acta Automatica Sinica*, vol. 29, no. 3, pp. 434–449, 2003.

[14] J. M. Ferryman, S. J. Maybank, and A. D. Worrall, "Visual surveillance for moving vehicles," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 187–197, 2000.

[15] http://www.detec.no.

[16] http://www.gotchanow.com.

[17] http://secure30.softcomca.com/fge_biz.

[18] T. Brodsky, R. Cohen, E. Cohen-Solal, et al., "Visual surveillance in retail stores and in the home," in *Advanced Video-based Surveillance Systems*, chapter 4, pp. 50–61, Kluwer Academic, Boston, Mass, USA, 2001.

[19] R. Cucchiara, C. Grana, A. Prati, G. Tardini, and R. Vezzani, "Using computer vision techniques for dangerous situation detection in domotic applications," in *Proceedings of the IEE Workshop on Intelligent Distributed Surveillance Systems (IDSS '04)*, pp. 1–5, London, UK, February 2004.

[20] D. Greenhill, P. Remagnino, and G. A. Jones, "VIGILANT: content querying of video surveillance streams," in *Video-Based Surveillance Systems*, P. Remagnino, G. A. Jones, N. Paragios, and C. S. Regazzoni, Eds., pp. 193–205, Kluwer Academic, Boston, Mass, USA, 2002.

[21] C. Micheloni, G. L. Foresti, and L. Snidaro, "A co-operative multicamera system for video-surveillance of parking lots," in *Proceedings of the IEE Workshop on Intelligent Distributed Surveillance Systems (IDSS '03)*, pp. 21–24, London, UK, February 2003.

[22] T. E. Boult, R. J. Micheals, X. Gao, and M. Eckmann, "Into the woods: visual surveillance of noncooperative and camouflaged targets in complex outdoor settings," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1382–1401, 2001.

[23] M. Xu, L. Lowey, and J. Orwell, "Architecture and algorithms for tracking football players with multiple cameras," in *Proceedings of the IEE Workshop on Intelligent Distributed Surveillance Systems (IDSS '04)*, pp. 51–56, London, UK, February 2004.

[24] J. Krumm, S. Harris, B. Meyers, B. Brumit, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easy living," in *Proceedings of 3rd IEEE International Workshop on Visual Surveillance (VS '00)*, pp. 3–10, Dublin, Ireland, July 2000.

[25] J. Heikkilä and O. Silvén, "A real-time system for monitoring of cyclists and pedestrians," in *Proceedings of 2nd IEEE International Workshop on Visual Surveillance (VS '99)*, pp. 74–81, Fort Collins, Colo, USA, 1999.

[26] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.

[27] http://www.cieffe.com.

[28] http://www.objectvideo.com.

[29] http://www.nice.com.

[30] http://www.pi-vision.com.

[31] http://www.ipsotek.com.

[32] http://www.neurodynamics.com.

[33] I. Haritaoglu, D. Harwood, and L. S. Davis, "Hydra: multiple people detection and tracking using silhouettes," in *Proceedings of 2nd IEEE Workshop on Visual Surveillance (VS '99)*, pp. 6–13, Fort Collins, Colo, USA, July 1999.

[34] J. Batista, P. Peixoto, and H. Araujo, "Real-time active visual surveillance by integrating peripheral motion detection with foveated tracking," in *Proceedings of the IEEE Workshop on Visual Surveillance (VS '98)*, pp. 18–25, Bombay, India, January 1998.

[35] Y. Ivanov, A. Bobick, and J. Liu, "Fast lighting independent background subtraction," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 199–207, 2000.

[36] R. Pless, T. Brodsky, and Y. Aloimonos, "Detecting independent motion: the statistics of temporal continuity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 768–773, 2000.

[37] L.-C. Liu, J.-C. Chien, H. Y.-H. Chuang, and C. C. Li, "A frame-level FSBM motion estimation architecture with large search range," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS '03)*, pp. 327–333, Miami, Fla, USA, July 2003.

[38] P. Remagnino, A. Baumberg, T. Grove, et al., "An integrated traffic and pedestrian model-based vision system," in *Proceedings of the British Machine Vision Conference (BMVC '97)*, pp. 380–389, Essex, UK, 1997.

[39] K. C. Ng, H. Ishiguro, M. Trivedi, and T. Sogo, "Monitoring dynamically changing environments by ubiquitous vision system," in *Proceedings of 2nd IEEE Workshop on Visual Surveillance (VS '99)*, pp. 67–74, Fort Collins, Colo, USA, July 1999.

[40] J. Orwell, P. Remagnino, and G. A. Jones, "Multi-camera colour tracking," in *Proceedings of the 2nd IEEE Workshop on Visual Surveillance (VS '99)*, pp. 14–21, Fort Collins, Colo, USA, July 1999.

[41] T. Darrell, G. Gordon, J. Woodfill, H. Baker, and M. Harville, "Robust real-time people tracking in open environments using integrated stereo, color, and face detection," in *Proceedings of the 3rd IEEE Workshop on Visual Surveillance (VS '98)*, pp. 26–33, Bombay, India, January 1998.

[42] N. Rota and M. Thonnat, "Video sequence interpretation for visual surveillance," in *Proceedings of 3rd IEEE International Workshop on Visual Surveillance (VS '00)*, pp. 59–68, Dublin, Ireland, July 2000.

[43] J. Owens and A. Hunter, "Application of the self-organising map to trajectory classification," in *Proceedings of 3rd IEEE International Workshop on Visual Surveillance (VS '00)*, pp. 77–83, Dublin, Ireland, July 2000.

[44] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[45] E. Stringa and C. S. Regazzoni, "Content-based retrieval and real time detection from video sequences acquired by surveillance systems," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '98)*, vol. 3, pp. 138–142, Chicago, Ill, USA, October 1998.

[46] C. Decleir, M.-S. Hacid, and J. Koulourndijan, "A database approach for modeling and querying video data," in *Proceedings of the 15th International Conference on Data Engineering (ICDE '99)*, pp. 1–22, Sydney, Australia, March 1999.

[47] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 205–210, Washington, DC, USA, June-July 2004.

[48] J. Black, T. Ellis, and P. Rosin, "A novel method for video tracking performance evaluation," in *Proceedings of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 125–132, Nice, France, October 2003.

[49] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: a review," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 152, no. 2, pp. 192–204, 2005.

[50] Y. Shoham, "Agent-oriented programming," *Artificial Intelligence*, vol. 60, no. 1, pp. 51–92, 1993.

[51] A. Newell, *Unified Theories of Cognition*, Harvard University Press, Cambridge, Mass, USA, 1990.

[52] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[53] Y. Labrou, T. Finin, and Y. Peng, "Agent communication languages: the current landscape," *IEEE Intelligent Systems & Their Applications*, vol. 14, no. 2, pp. 45–52, 1999.

[54] A. Rao and M. Georgeff, "BDI agents: from theory to practice," in *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS '95)*, V. Lesser, Ed., pp. 312–319, The MIT Press, San Francisco, Calif, USA, June 1995.

[55] A. Pokahr, L. Braubach, and W. Lamersdorf, "Jadex: implementing a BDI-infrastructure for JADE agents," in *EXP - In Search of Innovation (Special Issue on JADE)*, vol. 3, no. 3, pp. 76–85, September 2003, http://vsis-www.informatik.uni-hamburg.de/projects/jadex/.

[56] F. Bellifemine, A. Poggi, and G. Rimassa, "Developing multi-agent systems with JADE," in *Proceedings of the 7th International Workshop Agent Theories Architectures and Languages (ATAL '00)*, pp. 89–103, Boston, Mass, USA, July 2000, http://jade.tilab.com/.

[57] N. T. Nguyen, S. Venkatesh, G. West, and H. H. Bui, "Multiple camera coordination in a surveillance system," *Acta Automatica Sinica*, vol. 29, no. 3, pp. 408–422, 2003.

[58] H. H. Bui, S. Venkatesh, and G. A. W. West, "Tracking and surveillance in wide-area spatial environments using the abstract hidden Markov model," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 1, pp. 177–196, 2001.

[59] T. M. Rath and R. Manmatha, "Features for word spotting in historical manuscripts," in *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR '03)*, vol. 1, pp. 218–222, Edinburgh, Scotland, August 2003.

[60] T. Oates, M. D. Schmill, and P. R. Cohen, "A method for clustering the experiences of a mobile robot that accords with human judgments," in *Proceedings of the 7th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, pp. 846–851, AAAI Press, Austin, Tex, USA, July-August 2000.

[61] N. T. Nguyen, H. H. Bui, S. Venkatesh, and G. West, "Recognising and monitoring high-level behaviours in complex spatial environments," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 2, pp. 620–625, Madison, Wis, USA, June 2003.

[62] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852–872, 2000.

[63] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp, "Urban surveillance systems: from the laboratory to the commercial world," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1478–1497, 2001.

[64] R. T. Collins, A. J. Lipton, T. Kanade, et al., "A system for video surveillance and monitoring," Tech. Rep. CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pa, USA, 2000.

[65] C. S. Regazzoni, V. Ramesh, and G. L. Foresti, "Special issue on video communications, processing, and understanding for third generation surveillance systems," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1355–1539, 2001.

[66] M. Christensen and R. Alblas, "V2- design issues in distributed video surveillance systems," Tech. Rep., Department of Computer Science, Aalborg University, Aalborg, Denmark, 2000.

[67] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1456–1477, 2001.

[68] X. Yuan, Z. Sun, Y. Varol, and G. Bebis, "A distributed visual surveillance system," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS '03)*, pp. 199–205, Miami, Fla, USA, July 2003.

[69] I. Pavlidis and V. Morellas, "Two examples of indoor and outdoor surveillance systems," in *Video Based Surveillance Systems: Computer Vision and Distributed Processing*, P. Remagnino, J. A. Graeme, N. Paragios, and C. Regazzoni, Eds., pp. 39–50, Kluwer Academic, Boston, Mass, USA, 2002.

[70] J. A. Besada, J. García, J. Portillo, J. M. Molina, A. Varona, and G. Gonzalez, "Airport surface surveillance based on video images," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 1075–1082, 2005.

[71] J. García, J. M. Molina, J. A. Besada, and J. I. Portillo, "A multitarget tracking video system based on fuzzy and neuro-fuzzy techniques," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 14, pp. 2341–2358, 2005, special issue on Advances in Intelligent Vision Systems: Methods and Applications.

[72] J. García, O. Pérez, A. Berlanga, and J. M. Molina, "An evaluation metric for adjusting parameters of surveillance video systems," in *Focus on Robotics and Intelligent Systems Research*, Nova Science, New York, NY, USA, 2004.

[73] O. Pérez, J. García, A. Berlanga, and J. M. Molina, "Evolving parameters of surveillance video systems for non-overfitted learning," in *Proceedings of the 7th European Workshop on Evolutionary Computation in Image Analysis and Signal Processing (EvoIASP '05)*, pp. 386–395, Lausanne, Switzerland, March-April 2005.

[74] P. K. Varshney and I. L. Coman, "Distributed multi-sensor surveillance: issues and recent advances," in *Video-Based Surveillance Systems. Computer Vision and Distributed Processing*, pp. 239–250, Kluwer Academic, Boston, Mass, USA, 2002.

[75] L. Marchesotti, S. Piva, and C. Regazzoni, "An agent-based approach for tracking people in indoor complex environments," in *Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP '03)*, pp. 99–102, Mantova, Italy, September 2003.

[76] M. Bratman, *Intention, Plans, and Practical Reason*, Harvard University Press, Cambridge, Mass, USA, 1987.

[77] L. Braubach, A. Pokahr, D. Moldt, and W. Lamersdorf, "Goal representation for BDI agent systems," in *Proceedings of the 2nd Workshop on Programming Multiagent Systems: Languages, Frameworks, Techniques, and Tools (ProMAS '04)*, New York, NY, USA, July 2004.

[78] J. Carbó, A. Orfila, and A. Ribagorda, "Adaptive agents applied to intrusion detection," in *Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS '03)*, vol. 2691 of *Lecture Notes in Artificial Intelligence*, pp. 445–453, Springer, Prague, Czech Republic, June 2003.

M. A. Patricio received his B.S. degree in computer science from the Universidad Politécnica de Madrid in 1991, his M.S. degree in computer science in 1995, and his Ph.D. degree in artificial intelligence from the same university in 2002. He has held an administrative position at the Computer Science Department of the Universidad Politécnica de Madrid since 1993. He is currently Associate Professor at the Escuela Politécnica Superior of the Universidad Carlos III de Madrid and Research Fellow of the Applied Artificial Intelligence Group (GIAA). He has carried out a number of research projects and consulting activities in the areas of automatic visual inspection systems, texture recognition, neural networks, and industrial applications.

J. Carbó is an Associate Professor in the Computer Science Department of the Universidad Carlos III de Madrid. He is currently a Member of the Applied Artificial Intelligence Group (GIAA). Previously he belonged to other AI research groups at Universite de Savoie (France) and Universidad Politécnica de Madrid. He received his Ph.D. degree from the Universidad Carlos III in 2002 and his B.S. and M.S. degrees in computer science from University Politécnica de Madrid in 1997. He has over 10 publications in international journals and 20 in international conferences. He has organized several workshops and special sessions, and has acted as reviewer for several national and international conferences. He has researched on European funded projects (2 ESPRITS), a United Nations funded project (U.N.L.), and other national research initiatives. His interests focus on trust and reputation of agents, automated negotiations, fuzzy applications, and security issues of agents.

O. Pérez is currently a Research Assistant at Universidad Carlos III de Madrid, working with the Computer Science Department since year 2004. He received his degree in telecommunications engineering from Universidad Politécnica de Madrid in 2003. He is now working for the Applied Artificial Intelligence Research Group. His main interests are artificial intelligence applied to image data processing, video surveillance automatic systems, and computer vision.

J. García is currently an Associate Professor at Universidad Carlos III de Madrid, working for the Computer Science Department since year 2000. He received a degree in telecommunications engineering from Universidad Politécnica de Madrid in 1996, and his Ph.D. degree from the same university in 2001. He is now working for the Applied Artificial Intelligence Research Group. Before, he was a Member of the Data Processing and Simulation Group at Universidad Politécnica de Madrid. He has participated in several national and European projects related to air traffic management. His main interests are artificial intelligence applied to engineering aspects in the context of radar and image data processing, navigation, and air traffic management. He has authored more than 10 publications in journals and 30 in international conferences

J. M. Molina is an Associate Professor at the Universidad Carlos III de Madrid. He joined the Computer Science Department of the Universidad Carlos III de Madrid in 1993. Currently he coordinates the Applied Artificial Intelligence Group (GIAA). His current research focuses on the application of soft computing techniques (NN, evolutionary computation, fuzzy logic, and multiagent systems) to radar data processing, air traffic management, and e-commerce. He is the author of up to 20 journal papers and 80 conference papers. He received a degree in telecommunications engineering from the Universidad Politécnica de Madrid in 1993 and a Ph.D. degree from the same university in 1997.