A multi-agent architecture for supporting distributed normality-based intelligent surveillance

D. Vallejo^{*}, J. Albusac, J.J. Castro-Schez, C. Glez-Morcillo, L. Jiménez

School of Computer Science, Paseo de la Universidad, 4 - 13071 Ciudad Real, Spain

ARTICLE INFO

Article history:

Received 11 August 2009

Received in revised form

5 November 2010

Accepted 11 November 2010

Available online 30 November 2010

Keywords:

Distributed Intelligent Surveillance

Complex Monitoring

Multi-Agent Architecture

Knowledge Acquisition

Machine Learning

ABSTRACT

Intelligent surveillance involves the use of AI techniques to monitor environments whose analysis is becoming more and more complex because of the large number of sensors used and the need of monitoring multiple events of interest simultaneously. Most of the current surveillance systems provide solutions for particular problems but still suffer from lack of flexibility and scalability when they are used on different or related surveillance problems. To overcome this limitation, two aspects should be addressed: a knowledge-based surveillance model flexible enough to deal with different events of interest and an architecture that gives support to this model when deploying the surveillance system within a particular scenario. This paper discusses the architecture devised to deploy intelligent surveillance systems by means of a set of autonomous agents that are responsible for the management of different surveillance tasks and for cooperating to monitor complex environments. This multi-agent architecture is inspired by a normality-based formal model used to define the knowledge needed to analyze general-purpose surveillance concepts. We use the architecture to deploy a surveillance system to monitor an urban traffic scenario.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Intelligent surveillance can be defined as the application of techniques, algorithms, and methods with the aim of developing advanced security systems that perform the tasks commonly done by human operators. Currently, two of the key challenges of intelligent surveillance systems are the integration and interpretation of huge quantities of data collected by surveillance devices, typically video cameras, and the efficient application of expert knowledge-based models to provide advanced surveillance. Within this context, the main goal to reach consists in advancing the state of the art of traditional surveillance systems, in which the security staff continuously monitor an environment through CCTVs (close-circuit televisions) (Velastin and Remagnino, 2005). The main problems of this human dependence are the fatigue and the tiredness derived from the continuous monitoring of this kind of surveillance devices (Smith, 2004), as well as the difficulty of dealing with simultaneous threats when a chaotic situation takes place. Furthermore, the increase in the number of surveillance sensors implies that a security guard has to deal with more devices or, alternatively, to employ more human resources. In this field, the use of intelligent techniques may contribute to solve this kind of problems and to reduce the cost associated to the security staff by

semi-automating the tasks that are currently performed by them (Collins et al., 2000). Nowadays, there are non-commercial artificial systems able to infer anomalous behavior (Haritaoglu et al., 2000), to identify suspicious or lost objects (Stringa and Regazzoni, 2000), to analyze object movements (Hu et al., 2004), to recognize the gait of people (Tao et al., 2007; Xu et al., 2006), to detect crowds (Velastin et al., 2005), or to retrieve relevant information (Saykol et al., 2010) among other abilities (see Valera and Velastin, 2005 for a detailed revision).

Normally, surveillance systems are based on multi-layered architectures so that low-level layers are responsible for processing the data collected from the environment and high-level layers are in charge of performing the intelligent analysis of the scene, which may involve tasks such as decision support in response to dangerous situations. Fig. 1 shows the abstract architecture of a multi-layered surveillance system, where the more common stages and the main tasks that compose them are exposed.

Recently, the evolution of surveillance systems has been determined by two fundamental factors (Valera and Velastin, 2005): (i) the spatial distribution of information sources (such as video cameras, microphones, or infrared sensors), fact that implies the use of distributed processing and information fusion mechanisms, and (ii) the society demands for reaching a higher security both in public and private environments, which requires the use of more effective Computer Vision techniques (Forsyth and Ponce, 2002) and Artificial Intelligence methods to get a deeper knowledge from the monitored environment. In this way, the current trend when conceiving and

^{*} Corresponding author. Tel.: +34 926 295 300x6494; fax: +34 926 295 354.
E-mail address: David.Vallejo@uclm.es (D. Vallejo).

Fig. 1. Abstract architecture of an intelligent surveillance system.

advanced surveillance system consists in facing the design of knowledge-based distributed systems in which the environment information is used to generate knowledge and this knowledge is used to provide an advanced surveillance.

So far, the main research line is to give solutions to specific problems in particular environments. These approximations offer intelligent systems that get good results when monitoring an environment but they lack flexibility and scalability when deploying them in more general monitored environments or different from the initially conceived ones, which require the monitoring of a larger quantity of moving objects or the possibility of analyzing different surveillance questions. To face these limitations, two main issues need to be addressed:

- To adopt a cognitive surveillance system which is scalable and independent from the application domain to allow the deployment of analysis modules, being inspired by aspects such as trajectories, suspicious objects, etc. This adoption is justified by the need to monitor different threats, which generally depend on the environment.
- To design and develop a flexible architecture that gives support to this model, specifying those components needed to provide an integral solution and those mechanisms required (communication, configuration, deployment, etc.) to deploy the intelligent surveillance system.

This work addresses this second issue, taking into account the surveillance model based on normality analysis modules proposed in Albusac et al. (2009a). This architecture allows to create an intelligent and distributed security center to facilitate the integration of surveillance devices and the association of security experts around concepts, which refer to aspects or threads to monitor. Our proposal is based on the concept of intelligent agents (Wooldridge and Jennings, 1995), which can be defined as software programs that are able to autonomously develop tasks to reach particular aims. Within this context, the agents provide the mechanisms needed to perform the different tasks involved in a surveillance system and in a security control center. Furthermore, they represent the mechanisms used to emulate the behavior of human operators.

The rest of the paper is structured as follows. Section 2 justifies the multi-agent architecture proposed as the infrastructure needed to deploy intelligent surveillance systems. Section 3 briefly describes the theoretical model supported by the proposed architecture. In Section 4, the characteristics of the architecture are described in detail. Section 5 overviews how to deploy intelligent surveillance systems and goes in depth into the surveillance process. In Section 6, the deployment of a surveillance system in a particular urban traffic environment is studied. Finally, Section 7 resumes the work and suggests future research lines.

2. Related work

2.1. Specific architectures for surveillance

As previously stated, most architectures for surveillance are structured in layers so that each layer performs a well-defined task and offers a set of services to the rest of them. Within this context, a possible approach consists in using a set of computers to integrate multiple surveillance techniques such as in Haritaoglu et al. (2000). From an architectural point of view, the authors devised a first layer to detect moving objects and a second one to classify them. This system was conceived to detect silhouettes in order to analyze behaviors in outdoor environments. However, the architecture did not provide the mechanisms required to extend the system functionality.

Currently, PRISMATICA (Velastin et al., 2005) represents one of the most sophisticated surveillance systems, developed to detect relevant situations in complex environments. The architecture is composed of a set of devices which perform local monitoring on a reduced area. The information obtained can be later sent to a central manager. The authors conclude their work by discussing the need of integrating expert knowledge and automatic learning to provide more advanced surveillance.

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

A more recent work was proposed by Detmold et al. (2008), where they discussed a middleware as a mechanism to deploy intelligent video surveillance systems based on topologies that describe the activity of monitored objects. One of the innovative contributions is the use of a distributed blackboard to increase the flexibility and scalability of the communication model. Within this same research line, IBM presented an alternative (Tian et al., 2008) based on the intelligent management of the data gathered by the security devices and on open standards.

2.2. Multi-agents in intelligent surveillance

Surveillance environments present a dynamic, heterogeneous and complex nature where multiple tasks must be performed and their output must be merged to provide an advanced analysis of object behaviors. So, Multi-Agent Systems can be used to offer a natural solution to devise systems where intelligent agents communicate and cooperate to monitor complex environments. Next, the use of agent technology in multiple surveillance-related tasks is discussed.

One of the first stages of surveillance is segmentation (Forsyth and Ponce, 2002), which can be understood as the process that allows to distinguish objects of interest from visual information (e.g. single frames). There exist multiple proposals where agents have been used, such as the works discussed in Bovenkamp et al. (2004) and Lee (2003), to identify faces and adapt the segmentation process to the monitoring context.

The tracking process (Forsyth and Ponce, 2002), whose main goal is to track individual objects, has been one of the most addressed when using agents to monitor objects. Within this context, a direct approach consists in deploying one agent per identified object in order to analyze its behavior. Thus, it is possible for agents to communicate and cooperate to monitor multiple objects simultaneously. A representative example of this approach was proposed by Remagnino et al. (2004), where they design the camera agent to calibrate the camera, track objects and learning their behavior. Similar proposals were later discussed in Garcia et al. (2005), but paying special attention to the communication messages exchanged between agents; and Patricio (2007), where the authors deployed camera agents based on the belief–desire–intention model.

On top of the segmentation and tracking processes it is possible to deploy high-level modules to analyze behaviors and identify events of interest. Some of the previously referred works not only track objects but also perform some kind of intelligent analysis. For instance, in Remagnino et al. (2004) the authors design the object agent to identify behaviors depending on a set of specific activities. These agents can interact to detect whether two or more of them are monitoring a common moving object. Abreu et al. (2000) also devised a multi-layered architecture with multiple agents to track and classify behaviors.

One of the hot research areas in the context of behavior analysis is vehicle monitoring (Chen et al., 2009; Vallejo et al., 2009) in order to detect what vehicles behave incorrectly according to the traffic law.

Finally, information fusion techniques can be used to aggregate data from multiple sources in order to increase robustness and improve surveillance by combining the knowledge inferred by software agents. In Karlsson et al. (2005), the agents that monitor objects of interest use information fusion of data gathered by multiple sensors. Related works use this technique to detect inconsistencies and improve the monitoring results (Castanedo et al., 2008; Remagnino et al., 2004).

2.3. Our approach

The multi-agent architecture for supporting distributed intelligent surveillance proposed in this work addresses the challenge posed in Velastin et al. (2005), where the authors stress the need of integrating expert knowledge and machine-learning techniques to improve surveillance. That is why the core of our approach is a knowledge-based formal model for normality analysis (Albusac et al., 2009a) that allows to deploy the named surveillance components to monitor behaviors.

The main advantage of adopting this component-based approach is that when a general component has been designed, it can be instantiated in many surveillance environments as needed. For instance, if a component for defining and analyzing normal trajectories in any environment has been designed, the instantiation of such component would imply to set the normal trajectories that are particular for the monitored environment by means of the definitions established by the component. Typically, each different component will require a method to get the knowledge needed to instantiate such component having taken into account the particular problem where it will be used. Within this context, knowledge acquisition tools or

machine-learning algorithms can be used to reach two main goals: (i) to create instances of components in particular environments and (ii) to debug the functionality of the components.

Using such a general surveillance model implies a high number of needs that must be covered by the architecture that gives support to this model. In order to make use of this model in particular problems, an architecture that allows to add and instantiate components depending on the requirements of the issues addressed by the surveillance system is needed. Furthermore, this architecture must face the division of the problem into sub-problems, which usually involves dealing with distributed information in an intelligent way within the global environment, managing the tasks that are performed in each layer and providing the communication mechanisms needed to integrate them.

Regarding previously mentioned works (Detmold et al., 2008; Enficiaud et al., 2006; Tian et al., 2008; Velastin et al., 2005) which are not based on agent technology, our architecture pursues the scalability at a knowledge level since it allows to deploy software agents that manage the knowledge bases used to monitor environments. That is to say, our architecture does not only scale when the number of surveillance devices is increased but also when new threads or events of interest must be analyzed. Compared to Detmold et al. (2008), we use multiple blackboards (one per sub-environment) instead of approaching the knowledge sharing by means of a distributed blackboard. Thus, we guarantee robustness by replicating each single blackboard when required.

One of the main architectural differences between our approach and the followed in agent-based surveillance architectures such as Garcia et al. (2005), Patricio (2007), and Remagnino et al. (2004) is that we deploy agents specialized in analyzing the normality of single surveillance concepts, according to the formal model discussed in the next section. This choice may be more suitable from a theoretical model because each agent manages the knowledge needed to analyze an event of interest in the whole scenario. From the implementation point of view, replication schemes can be directly used to distribute the workload of the agents that monitor aspects that require more computational resources. Furthermore, the proposed architecture considers different classes of agents depending on their goal (e.g. preprocessing agents, normality agents, abnormality agents, fusion information agents etc.), which increase flexibility when monitoring complex environments and facilitates the configuration of the deployed system.

3. Rational of the surveillance architecture: a model for normality analysis

One of the main goals of intelligent surveillance is to provide mechanisms for automating the different tasks that are involved in this complex process, as depicted in Fig. 1. Within this context, one of the hottest research topics is behavior understanding. To address this challenge, the normality of the monitored environment can be modeled or learned to define the usual situations that happen in it. However, it is also possible to model or learn the knowledge needed to represent the anomalous situations or possible threats that may take place in an environment.

This second approach may be suitable for detecting certain anomalies but lacks scalability when developing a surveillance system that aims to detect all the possible anomalous situations of the monitored environment because of the high number of them, some even unpredictable. In fact, this approach would imply to manage a large number of situations so that if one of them happens but was not modeled, then the system would be unable to detect it. On the contrary, an approach based on the normality definition has the advantage of being generally well known and delimited in most environments. In this way, a surveillance system based on this last

model would infer that a detected situation which was not explicitly modeled within the knowledge base is considered as abnormal or represents a possible threat. Furthermore, a normality-based approach can be easily complemented with the definition of the most common anomalous situations in order to be able to act according to certain threats.

This section briefly describes the highly scalable formal model that has inspired the multi-agent architecture devised in this paper. Such model allows to define and identify the normality of an environment according to different concepts (study of object trajectories, object speed, proximity relationships, etc.) by dealing with the uncertainty and imprecision existing in the data obtained from surveillance sensors (e.g. video cameras).

The proposed multi-agent architecture gives support to this normality analysis model, establishing a well-defined framework for the deployment of intelligent surveillance systems. Although this model is described in depth in Albusac et al. (2009a), this section summarizes its main concepts to facilitate the existing link between the model and the architecture and shows how the different agents that compose the architecture emerge in a natural way. These agents are responsible for supporting the needs imposed by the normality model when deploying an intelligent surveillance system into a particular problem.

Basically, the model deals with the problem of surveillance by dividing a physical environment into multiple sub-environments or perceptions ($P = \{E_1, E_2, \dots, E_i, \dots, E_n\}$). Then, in each sub-environment (E_i) the objects (e.g. pedestrians, vehicles, etc.) to monitor and the events or interest of threads (e.g. trajectories, speed, crowds etc.) to analyze must be established. To carry out such tasks, the information gathered by the deployed surveillance sensors is used. Commonly, the information obtained from the sensor must be processed in order to generate high-level information (e.g. video camera).

To face this second issue, the proposed architecture allows to deploy the named *preprocessing agents* (PA), which maintain a reactive model and use a set of techniques to process the information provided by one or more surveillance sensors and to generate a certain output. In the case of video surveillance, sensors are video cameras and the preprocessing agents would be in charge for the segmentation and tracking processes (Forsyth and Ponce, 2002) to identify and track moving objects, respectively.

These events of interest or threads, formally defined as normality concept (C) in Albusac et al. (2009a), determine the normality of monitored scenarios according to a certain object class. Such concepts, defined independently of any environment, are based on the named *normality constraints*, which must be satisfied for an object to behave in a normal way according to the concept c_i ($\Phi_i = \{\mu_{i1}, \mu_{i2}, \dots, \mu_{ki}\}$).

Some kinds of constraints are *role constraints*, which set the kind of objects that is allowed to perform a certain action; *spatial constraints*, which set where objects can be placed, their speed, the kind of movements that can do, etc.; or *temporal constraints*, which set when objects can perform a certain action.

After having defined the analyzed concepts and their respective constraints applied to a certain environment in a general way, these concepts must be instantiated for the different sub-environments to be monitored. For example, if normal trajectories are to be addressed, an instance of such concept would be the specific normal trajectories that can be done in a certain zone of the environment which is monitored by a set of video cameras.

In order to manage the knowledge of a concept in a certain sub-environment, the multi-agent architecture supports the deployment of the named *normality agents* (NA). These deliberative agents maintain a knowledge model of the sub-environment that they study and are capable of reasoning to infer if objects behave in a normal way according to the concept in which they are specialized

in. In this way, a normality agent na_i will be deployed for each concept $c_i \in C$ at a sub-environment level.

The next step consists of combining the normality values given for each surveillance concept that takes part in the normality analysis. The analysis made by this agent represents the degree of normality of the behavior of an object in the monitored environment. In order to combine the output from each surveillance concept, the normality model proposed in Albusac et al. (2009a) has been extended thanks to the use of OWA operators (Yager, 1988). This kind of operators allows to make decisions taking into account many criteria so that each criterion has a weight that directly affects to the final decision. In this case, the output of each surveillance concept is a criterion and the global normality value corresponds with this final decision.

To deal with the normality analysis based on normality concepts in a whole sub-environment, the multi-agent architecture allows the deployment of an *environmental normality analysis agent* for each sub-environment E_i that forms the global environment, which is responsible for checking if every monitored object *obj* satisfies the normality associated to the concepts that are monitored in such sub-environment.

Although this model is based on the definition of normal situations, it is generally desirable to know or predict the most common anomalous or dangerous situations that can take place in an environment to act accordingly. To deal with this need, the architecture proposed in this work complements the discussed normality model with the named *abnormality agents* (AA). These agents are responsible for managing the knowledge that specifies the most common anomalous situations of a particular concept and able to infer them thanks to the knowledge generated by the normality agents, which deal with the same concept and their own knowledge. For instance, if a normality agent na_i detects that a vehicle did not satisfy the normality of the concept trajectories, then the related abnormality agent aa_i may use this information to infer that the particular anomalous situation was due to the invasion of a space reserved for pedestrians, for example.

Therefore, one-to-one relationships are established between normality and abnormality agents according to a certain concept within a sub-environment. Such links require a more direct communication mechanism than the events-based one. This is why the architecture also supports searches by name (white-pages service) and by service provided (yellow-pages service) as well as the sending of messages between agents (message transport service). These services have been widely used in the Multi-Agent Systems field. In this work, they have been modeled according to the specifications of the FIPA (Foundation for Intelligent Physical Agents, FIPA, 2004) committee.

Currently, the normality model adopted supports a local reasoning model at sub-environment E_i level, which has been stated specifically in the architecture with the environmental normality analysis agent. The next step consists in making a reasoning at global level. Within this context, a fusion module that is able to aggregate the knowledge generated by the different normality agents that carry out the analysis of a same concept in related (commonly in terms of physical proximity) sub-environments is needed. In view of this requirement, the architecture defines a fusion information module that brings together those agents responsible for analyzing the normality of a concept at a global level (*fusion information agents*), an agent in charge of deciding when the information fusion is needed (*fusion manager agent*), or an agent that allows to establish links between the knowledge generated by these global normality agents (*global fusion agent*).

The interactions that emerge from these agents can be tackled with a communication model based on the *blackboard architecture* (Englemore and Morgan, 1988). In this way, the normality agents can publish the partial knowledge needed by the fusion information agents to perform the global analysis of the scene. Furthermore, the

architecture allows to deploy as many fusion information agents as needed, taking into account that each one of them deals with the knowledge of a concept c_i from a global perspective.

Finally, and considering the abstract architecture of surveillance systems shown in Fig. 1, two of the more relevant high-level tasks are decision support and crisis management. Such responsibilities can be conceived by means of decision support modules that suggest actions to the security personnel in response to the detection of an anomalous or dangerous situation. This is why the architecture integrates the *decision support agent* as the agent that deals with these issues. This agent interacts with all the agents that participate in the surveillance process.

4. The multi-agent architecture for intelligent surveillance

4.1. Architecture overview

The needs raised by the normality model described in the previous section have been used as the base for designing the multi-agent architecture for the deployment of intelligent surveillance systems. Fig. 2 shows the architecture proposed in this paper to carry out this task. This architecture is structured into three levels: reactive level, deliberative level, and user level. The different components of each level communicate with one another via a communication middleware that allows the interaction between them and the integration of new components. Next, the main characteristics of each one of these levels are summarized:

- The *reactive level* is composed of the surveillance devices used to monitor the sub-environments of the global scenario and of the agents that maintain a reactive model.
- The *deliberative level* is composed of those agents that maintain and use a knowledge model in order to analyze the normality and the abnormality of the environment from different points of view (components), as well as those agents that aggregate the information obtained from the sub-environments that establish the whole scenario and make decisions.
- The *user level* allows the interaction with the security expert via monitoring tools.

Besides devising the agents required to monitor environments from a global point of view, it is also interesting to establish a set of management agents that facilitate the control of the multi-agent system and provide the rest of agents with basic services, such as a yellow-pages service or a communication service. In order to face this need, the architecture defines a set of management agents according to the set of standards defined by the FIPA committee (Foundation for Intelligent Physical Agents, FIPA, 2004) for the development of multi-agent systems. These agents are the *Agent Management System (AMS)*, which manages the multi-agent system and control the agents life cycle; the *Directory Facilitator (DF)*, which provides a yellow-pages service; and the *Message Transport Service (MTS)*, which deals with the sending and reception of messages exchanged between agents.

Next, the information flow between the components that compose the architecture is described (see Fig. 3) in order to give an overview of how the information gathered from the environment sensors is converted into the high-level knowledge used to analyze behaviors and make decisions.

1. *Gathering information from the environment*: Surveillance sensors get information from the monitored scenario, which is related to the kind of sensor (audio, video, presence, etc). In the case of

Fig. 2. Architecture proposed for intelligent surveillance systems.

Fig. 3. Overview of the information flow between the main components of the surveillance architecture.

video surveillance, this information is the sequence of static images obtained from the surveillance cameras. The integration of these devices into the system is done via the communication middleware that supports the proposed architecture.

2. *Notifying surveillance events:* The output of the sensors may require processing to extract the high-level information used to analyze the normality of the environment. For instance, the visual information from cameras must be processed by preprocessing agents before sending it to the upper level. However, simple devices, such as presence sensors, are possible to directly send their output to the upper level. In both cases, this information is encapsulated into surveillance events that will be notified via event channels to the deliberative level.
3. *Generating knowledge to analyze the environment:* Surveillance events from the reactive level are processed by normality agents to study the normality and detect possible normality violations. The knowledge generated by these agents is written on the blackboard of each particular sub-environment. This approach based on a distributed blackboard improves the scalability and robustness of the surveillance system. Thus, the knowledge written on the blackboard of each sub-environment is local but, at the same time, allows the fusion information agents to infer more detailed conclusions about the monitoring process. Furthermore, when a normality agent detects that the normality according to a particular aspect is not met, it sends a notification to the abnormality agent that manages the more common anomalous situations of the same aspect and to the security personnel via the monitoring tools. The knowledge shared by the normality agents of a same sub-environment is used by the environmental normality analysis agent to study the normality at a sub-environment level, that is, taking into account all the aspects monitored in the associated sub-environment.
4. *Fusing information:* The explicit knowledge generated in each sub-environment is used by the fusion information agents, which are responsible for analyzing the normality of each aspect

or threat from a global point of view. This analysis is carried out by making use of the partial knowledge obtained from the normality agents that monitor a same aspect in the different sub-environments that compose the global scenario. This global study typically requires a mechanism to translate local information to global information to relate the knowledge obtained from each sub-environment. Within this context, the *global translator agent* is responsible for this process in order to provide a homogeneous vision of the information needed by the fusion information agents. Finally, the global fusion agent aggregates the knowledge of the different fusion agents to provide a more sophisticated surveillance that relates several surveillance aspects.

5. *Supporting the decision-making process:* The knowledge generated by the agents of each sub-environment and the global normality analysis module is used by the decision support and crisis manager agent to suggest possible actions to the security personnel in response to an anomaly detection. These interactions are made via monitoring tools.

4.2. Communication mechanisms

This section describes the communication mechanisms provided by the multi-agent architecture in order to give support for the agents to communicate with one another in the process of intelligent surveillance. Fig. 4 graphically shows the three currently existing mechanisms: (a) event-based communication, (b) direct communication between agents, and (c) communication through a blackboard system.

Event-based communication (see Fig. 4a) is used to deploy typed event channels, that is, the communication revolves around the kind of information which is sent and received instead of taking into account the identity of senders or receivers. This class of communication perfectly fits in the interaction between the components of the reactive level (sensors and preprocessing agents) and the agents of the deliberative level (normality agents,

Fig. 4. Communication mechanisms supported by the architecture: (a) event-based, (b) between agents, (c) through a blackboard system.

among others). The key concept of this mechanism is the *event*, which represents the abstraction between the publishers and the subscribers of information.

Direct communication allows an agent to communicate with one or more agents whose physical address is explicitly established within the message to be sent. This class of communication can be done with no intermediary (agent–agent) or through an agent that implements the message transport system specified by FIPA (*message transport service*). This communication mechanism is suitable for certain situations, such as those that cover the interactions between a normality agent and an abnormality agent that analyze the same surveillance concept.

Finally, the communication through a *blackboard system* supported by the architecture makes possible that several agents share knowledge when monitoring an environment. Typically, the blackboard will be used to write results by the normality agents that analyze a common concept, e.g. trajectories, in two or more physically related sub-environments. Thanks to this mechanism, high-level agents such as the fusion agents can read this partial knowledge to provide a more effective surveillance by aggregating this information.

4.3. Systematic requirements

This section justifies how the devised architecture provides the set of systematic requirements that any surveillance system should support, paying special attention to the practical deployment of this kind of applications. To address this challenge, our point of reference is the work discussed in Detmold et al. (2008), where the authors proposed a middleware for video surveillance and stated a well-defined set of systematic requirements. The justification of this reference lies in the similarity between the two proposals (discussed in this paper and Detmold et al., 2008) and the authors' reputation in the field.¹

These requirements are as follows:

- *Scalability*, as the capacity and mechanisms provided by the architecture to integrate new components.
- *Availability*, related to the system robustness and the detection of failures and their associated consequences.
- *Evolvability*, which refers to the system response when making hardware or software changes.
- *Integration*, as the architecture capacity of integrating new devices.
- *Security*, as the provided mechanisms to fight against inadequate or unauthorized uses.
- *Manageability*, as the interaction between the security personnel and the software system.

Regarding scalability, the proposed architecture considers two issues. On the one hand, the provided support when integrating new surveillance sensors. On the other hand, the need of monitoring new events of interest or threads.

The integration of a new surveillance sensor is done in the reactive level, being a preprocessing agent required depending on the sensor class. Within this context, the scalability is guaranteed thanks to the event channels that connect the reactive and deliberative levels. The need of monitoring new events of interest is achieved thanks to the possibility of deploying normality/abnormality agents when required. This solution is linked to the integration requirement and will not be described later.

Availability has been managed by means of event channels and replication mechanisms. In this way, if a non-replicated agent that analyzes a concept terminates abruptly, then its subscribers will keep on working but without the knowledge previously inferred by such an agent.

Regarding the security mechanisms provided by the architecture, multiple methods have been considered to protect the information, ensure its integrity or verify the identity of the agents.

¹ <http://www.acvt.com.au/research/surveillance/>

Within this context, the implemented prototype supports the SSL protocol. Finally, in terms of manageability, the architecture pays special attention to the deployment process. Thus, agent factories are provided to automate the agent deployment.

5. Using the multi-agent architecture to deploy intelligent surveillance systems and monitor complex environments

From an abstract point of view and considering the normality analysis model described in Section 3, the design of surveillance systems based on the analysis of concepts implies the management of the following issues:

1. Identify the set of sensors S required and/or available in the environment and the mechanisms to process the information gathered by them when needed.
2. Establish the division of the global environment into a set of perceptions or sub-environments $P = \{E_1, E_2, \dots, E_n\}$ with the aim to simplify the surveillance.
3. Determine the moving objects O to monitor or, in a general way, the environment elements to pay attention.
4. Design or reuse the normality components for the analysis of concepts C to be monitored within the environment.

This set of general steps has a direct reflect on the proposed architecture regarding the three level structure. At a reactive level, both the sensors used and the preprocessing agents deployed are needed to be integrated. At a deliberative level, the normality agents responsible for the analysis of each concept $c_i \in C$ are needed to be deployed, as well as the communication mechanisms required for the agents to communicate and cooperate with one another. Finally, at a user level, the monitoring and knowledge acquisition tools will interact with the agents that monitor the scenario.

5.1. Deployment and configuration of surveillance agents

At a *reactive level*, two groups of agents are distinguished: (i) those responsible for processing the information gathered by the set of sensors S deployed in the environment, that is, the preprocessing agents PA and (ii) those responsible for managing the basic operations of any domain-independent multi-agent system, that is, $MA = \{pm, sm, cm\}$, recalling that pm represents the manager of the agent platform, sm is the service manager or yellow-pages agent, and cm is the communication manager agent.

In each sub-environment E_i , a preprocessing agent pa_j for each kind of sensor existing within E_i that requires an information processing will be deployed, such as video or audio sensors. For instance, if a sub-environment E_1 is composed of a video camera s_1 , a microphone s_2 , and a presence sensor s_3 , then a video preprocessing agent pa_1 and an audio preprocessing agent pa_2 will be typically deployed. These agents would analyze the information obtained from the sensors s_1 and s_2 , respectively. In this case, s_3 would not require the deployment of another preprocessing agent since its output is expressed in boolean terms, which does not need an extra processing before notifying to the deliberative level. The rest of sub-environments that compose the global environment are dealt in a similar way.

At a deliberative level, a distinction can be made depending on the scope of the analysis carried out by the agents that compose this level between a local scope at a sub-environment E_i level or global scope at an environment level. Thus, the normality agents NAA , the abnormality agents AAA , and the environmental analysis agents $ENAA$ have a local scope from the sub-environment perspective,

while the agents that compose the information fusion module $GNAM$ and the decision-making module DSM have a global scope.

In this way, a normality agent na_j will be deployed mandatory in each sub-environment E_i for each concept c_k to be monitored in E_i . For example, if the concepts c_1 trajectories of moving objects and c_2 normal behavior in a pedestrian crossing are analyzed in a sub-environment E_i , then two normality agents na_1 and na_2 will be deployed for dealing with c_1 and c_2 , respectively. Optionally, an abnormality agent aa_j could be instantiated for each na_j in order to complement the normality analysis of na_j . Thus, when na_j detects a situation that does not satisfy the normality definition of c_j in E_i , it will communicate with aa_j for this abnormality agent to try to detect the anomalous situation that took place.

An example of surveillance concept to monitor the normality of trajectories, independently of the monitored environment, is discussed in depth in Albusac et al. (2009a). In this particular case, the knowledge base of the agent that monitors the trajectories in a particular urban environment, such as the one in Section 6.1, is populated by using a graphical knowledge acquisition tool. On the other hand, the use of machine-learning algorithms is discussed in Albusac et al. (2009b), paying special attention to the concept *normal speed* and how to generate the knowledge base for a normality agent to automatically integrate it.

Furthermore, a single environmental normality analysis agent ena_i will be deployed in each sub-environment E_i , which is responsible for checking if every object obj satisfies the normality definition according to the concepts that are being monitored in E_i . The logic followed by this agent, which aggregates the knowledge written by the normality agents on the blackboard, is discussed in depth in Albusac et al. (2009a).

In order to perform the global surveillance of the environment, the agents needed to fuse or aggregate the knowledge inferred by the previously mentioned agents are needed to be instantiated. This process implies the deployment of the following agents:

1. An agent ta to translate the local information managed by $ENAA$ into global information that can be used by the agents of the module $GNAM$ (see Fig. 2).
2. An agent fma to decide if the knowledge generated by the normality agents has to be aggregated and what fusion agents will be in charge of that.
3. A fusion information agent fa_i for each one of the concepts c_i studied in each sub-environment E_j . The more typical case is to fuse the information related to a single object obj obtained by two normality agents na_k and na_l of different sub-environments that deal with the same concept c_i , such as trajectories.
4. A fusion global agent ga to aggregate the knowledge generated by the fusion agents FA , the normality agents NAA , the environmental normality analysis agents $ENNA$, and the abnormality agents AAA .

Finally, a single decision support agent $dssa$ will be optionally deployed to act as a mediator between the surveillance system and the security personnel who use the monitoring tools and manage the alarms of the environment.

5.2. Deployment of the communications system

In the reactive level (see Fig. 2), the kind of communication is based on events and event channels. The latter represent the glue between the reactive level and the deliberative level. In this way, the information gathered by the sensors S and the preprocessing agents PA is the input flow of the event channels, while the normality analysis agents NAA of the deliberative level are the receivers (subscribers) of this information. Depending on the kind

Fig. 5. Sequence diagram in AUM notation for the analysis of a sub-environment E_i .

of information obtained in each sub-environment E_i that forms part of the global environment, one or more event channels can be deployed in each E_i . However, the more common practice will consist in deploying a single event channel for each kind of information (visual, audible, sensory etc.) gathered from each E_i . In this same level, one or more instances of the agent *cm* (message transport service) will be deployed to support direct communication between agents.

In the deliberative level (see Fig. 2), the communication between a normality agent na_i and an abnormality agent aa_i that monitor a same aspect or concept c_i has to be first addressed. In this case, na_i needs to know the existence of aa_i , information that can obtain by directly consulting the yellow-pages agent *sm* to obtain the physical address of aa_i . Now, na_i is able to communicate with aa_i in order to send the information required by aa_i to try to detect the anomaly that took place.

Within this same level, a blackboard system is deployed in each one of the sub-environments $E_1, E_2, \dots, E_n$ that compose the global environment. In this way, the normality agents NA_i and the abnormality agents AA_i of E_i share the knowledge generated in each step of the surveillance process through the blackboard. This common repository is used by the environmental analysis agent ena_i of each E_i to study the normality of the monitored objects according to the set of concepts $C_i \subseteq C$ analyzed in E_i .

Within the context of the global information fusion module GNA, a blackboard system where the fusion agents FA write the knowledge related to each concept c_i from a global point of view is deployed. On the other hand, the decision support and crisis manager agent *dssa* uses the different blackboards deployed in the deliberative level as well as the query modules of the rest of agents to perform its task.

In the user level there are two ways of communication: (i) the query module of the agents of the deliberative level and (ii) the notification of events on behalf of such agents to the monitoring tools deployed in the user layer. In the first case, the agents simply send the information requested by the security personnel, acting in a reactive way. In the second case, the agents notify the relevant information in an active way by using the event channels. For this

reason, the more common practice consists in deploying the following event channels:

- One for each sub-environment $E_i \in E$ where the abnormality agents AAA_i and the environmental normality analysis agent ena_i publish information.
- An event channel where the agents of the global normality analysis module GNA publish information.
- An event channel exclusively dedicated to the decision making and crisis manager agent *dssa* to give support to the security personnel when detecting a particular anomalous situation.

5.3. Analyzing monitored environments

The monitoring process can be understood as a loop of receiving events from the reactive layer, processing in the deliberative layer, and, finally, notifying the knowledge generated to the user layer via monitoring tools.

The first stage of the surveillance process consists in analyzing the normality and abnormality of each sub-environment E_i , using a subset of sensors $S_i \subseteq S$, a subset of preprocessing agents $PA_i \subseteq PA$, a subset of normality agents $NA_i \subseteq NA$, a subset of abnormality agents $AA_i \subseteq AA$, and an environmental normality analysis agent ena_i . Fig. 5 shows the interaction flow of a sub-environment E_i where each one of the mentioned agents participate. In this section, the particular case of video surveillance will be studied in order to detail the functionality of the agents that take part in the process of intelligent surveillance.

Under this hypothesis, the preprocessing agent pa_i receives the information gathered by a surveillance camera and carries out the processes of segmentation and tracking to determine the positions of monitored objects and how they are classified. This basic information, specially the kind of the moving object, reflects some degree of uncertainty due to the difficulty of determining it with absolute certainty. For example, pa_i may establish that a detected object labeled as obj_1 is on the position [320, 460], being 320 the

pixel that represents the position of obj_i in the x -axis and 460 the pixel in the y -axis. Furthermore, this object may belong to the class *car* with a membership value of 0.8 (being 1.0 the maximum) and to the class *pedestrian* with a value of 0.2, respectively.

Next, the preprocessing agent pa_i sends such information through an event channel for the subscribers, represented by the normality agent na_j in Fig. 5, to receive it. This information passing is denoted by means of the *analysis* operation of the diagram. Thanks to this information na_j is able to use its knowledge base $\tilde{KB}_j$ to infer how the monitoring of E_i progresses according to $\tilde{KB}_j$ and depending on the concept c_k .

The knowledge inferred by na_j is notified to the monitoring tools to allow the visualization of this kind of information, that is, the evolution of the surveillance process in each sub-environment, and written on the blackboard for ena_j to be able to carry out the global normality analysis of the sub-environment.

Finally, the normality analysis for the concept c_k studied by na_j can be complemented with the abnormality analysis done by aa_j for the same concept c_k . In this case, by means of the operation *notify_abnormality*, aa_j gets the information needed to identify the anomalous situation if it is specified in the knowledge base $\tilde{KB}_j$ of aa_j . This information comprises the satisfaction degrees of the normality constraints of all the instances of c_k associated to every monitored object in the last time of analysis in the sub-environment. This information is used by aa_j to detect if the behavior of any object that violated the normality definition of c_k matches with the one covered in its knowledge base.

The second stage of the surveillance process is defined as the fusion of the information obtained from the sub-environments $E_1, E_2, \dots, E_n$ that compose the global environment. This information is managed by the environmental normality analysis agents $ena_1, ena_2, \dots, ena_n$. Fig. 6 shows the interaction diagram that reflects how the agents that take part in the information fusion process communicate with one another to carry out the global surveillance of the environment.

Firstly, and due to each ena_k manages local information in E_k , the process of translating local information into global has to be carried out by the global translator agent gta . In the case of video surveillance, this process would imply, among other issues, translating the local

positions of an object obj in each E_j into a global position, only if the physical relations between sub-environments allow this process.

Next, each ena_k notifies the information of the normality analysis of E_k to the fusion manager agent fma , who is responsible for deciding whether the information fusion is required and, in that case, establishing what fusion agent is in charge of aggregating the knowledge inferred by the normality agents. In the case of video surveillance, this decision can be made depending on the physical relation between the monitored sub-environments, that is, depending on whether it is possible to relate the positioning information of a subset of sub-environments E_j , usually associated to the relations of physical adjacency between them.

As previously mentioned in Section 5.2, a blackboard system is deployed for each analyzed concept $c_k \in C$, which is used by normality agents of each sub-environment in order to share a common knowledge repository that reflects the progress of the surveillance process. In this way, the fusion information agent fa_i that manages the concept or threat c_k can read the blackboard and aggregate the knowledge inferred by the different local normality agents. Two main issues appear when facing this situation:

1. A mutual exclusion mechanism that ensures the right access to the blackboard is required.
2. The fusion information agents must include a mechanism that allows them to know when carrying out the information fusion process.

The first issue has been solved by using a monitor to ensure the right access to the blackboard. The second one directly depends on the concept c_k managed by the fusion agent and on the blackboard state so that the knowledge base $\tilde{KB}_i$ of each fa_i must address this issue. In the case of video surveillance, our solution is based on aggregating information as soon as the concept c_k allows, obtaining in this way global conclusions that are more precise when the blackboard contains more knowledge about the temporal evolution of the surveillance process (see Albusac et al., 2009a for a detailed discussion).

Finally, the architecture addresses the interaction between the fusion agents FA and the agent in charge of the global fusion ga that makes use of the knowledge individually inferred by the fusion

Fig. 6. Sequence diagram in AUML of the information fusion process.

agents. In the case of video surveillance, this global agent may support value-added services in case of emergency, where the normality associated to a concept c_k is altered by the exceptional situation. For example, if an anomalous situation, where there a fire is detected and, simultaneously, another anomalous situation consisting a dangerous driving on behalf of a vehicle close to the fire is also detected, then ga may be able to infer that, within this exceptional situation, the vehicle could be acting in a normal way because it could be running away from the fire or could be trying to help if it is an emergency vehicle.

All the processes of normality and abnormality analysis within an environment, local or global, end with the decision support and crisis manager agent $dssa$ who is responsible for suggesting actions to the security personnel through the monitoring tools. This agent, who must be understood as a collaborator in this process, uses the knowledge explicitly written in the blackboards and the query modules of the rest of agents to suggest possible actions in response to the detection of a dangerous situation. A simple implementation of this agent could be a rule-based system where the consequents of the rules are these possible actions expressed in a certain language.

6. Deploying the architecture in a particular surveillance problem

In this section, a solution to a surveillance problem in an urban traffic scenario where vehicles and pedestrians move around is discussed, using the ideas previously described, in order to illustrate our approach.

6.1. Environment description

The scenario chosen for deploying an intelligent surveillance system based on the model which is supported by the multi-agent

architecture proposed in this work (see Section 3) is the typical urban traffic environment shown in Fig. 7. This scenario consists a junction controlled by a roundabout where entering traffic must yield to traffic already in the circulatory roadway. All the lanes are one-way so that each branch of the roundabout has a single input and a single output. In this particular case, the global scenario (see Fig. 7a) has been divided into three sub-environments or perceptions: E_1 (see Fig. 7b), E_2 (see Fig. 7c), and E_3 (see Fig. 7d). The camera positions stem from those physical places that offer the best visibility and are easy to access from our research lab.

Currently, we are interested in carrying out the surveillance of the environment of Fig. 7 by means of the analysis of two independent concepts or threats. One is the trajectories of moving objects, denoted as c_1 from now on, and the other is the speed of such moving objects, denoted as c_2 from now on. To cover this surveillance requirements, these two concepts have been designed by using the definitions provided by the normality model. In this way, it is possible to instantiate them in each one of the sub-environments (E_1 , E_2 , and E_3) that compose the traffic scenario, which implies the deployment of the associated surveillance agents.

On the one hand, the analysis of trajectories aims to monitor the normality of moving objects (pedestrians and vehicles) when they move around the environment. The normality component of trajectories is discussed in Albusac et al. (2009a), where its design decisions and the constraints defined to establish its normality are discussed in depth.

For example, in the case of vehicles the normal behavior implies the definition of the possible trajectories that the monitored object can perform when it is moving towards the roundabout. To do that, a knowledge acquisition tool can be used to facilitate issues such as the division of the sub-environment where the component is instantiated into regions (see Fig. 8).

On the other hand, the analysis of moving objects speed aims to monitor their displacements in the environment. This is particularly

Fig. 7. Monitored urban traffic environment: (a) global vision of the scenario, (b) sub-environment E_1 monitored from the camera s_1 , (c) sub-environment E_2 monitored from the camera s_2 , (d) sub-environment E_3 monitored from the camera s_3 .

Fig. 8. (a) Image taken from the camera that monitors the sub-environment E_2 , (b) set of relevant zones defined in E_2 by an expert user.

Table 1
Instances deployed from the normality concepts of trajectories (c_1) and speed (c_2) monitored in E_1 , E_2 , and E_3 .

Concept	Sub-environment		
	E_1	E_2	E_3
Trajectories (c_1)	c_{1y}^1	c_{1y}^2	c_{1y}^3
Speed (c_2)		c_{2y}^2	c_{2y}^3

interesting in the case of vehicles not only when they approximate to the round about, but also regarding the pedestrian crossings to avoid knockdowns.

Table 1 summarizes the instances of the normality components c_1 and c_2 that are deployed in E_1 , E_2 , and E_3 . This specific deployment determines the particular architecture of the surveillance system devised for the case study discussed in this section. This particular multi-agent architecture, together with the communication mechanisms needed to provide a global surveillance, is the subject of the next subsection.

6.2. Discussion of the multi-agent system deployed

Fig. 9 graphically shows the particular architecture of the multi-agent intelligent surveillance system devised to monitor trajectories and pedestrian crossings in the scenario of Fig. 7. The process of intelligent surveillance has been carried out by dividing the scenario into three sub-environments (E_1 , E_2 , and E_3), which simplifies its global analysis and allows to compose the study of its different partial views.

In the reactive level the agents pm , sm , and cm are deployed to provide the basic management services, according to FIPA. A video camera, denoted as s_1 , is placed within the sub-environment E_1 . The output of this sensor is analyzed by the preprocessing agent pa_1 . In the sub-environments E_2 and E_3 , the cameras s_2 and s_3 and the preprocessing agents pa_2 and pa_3 are, respectively, deployed.

In the deliberative level, the first agents responsible for analyzing the scenario carry out the monitoring at a sub-environment level. In the sub-environment E_1 , the normality agent na_1 , which monitors the trajectories concept c_{1y}^1 , receives the input from the event channel deployed in E_1 . In E_1 the trajectories concept is the only one to be monitored since there is no pedestrian crossing that can be clearly perceived. An abnormality agent aa_1 has been deployed in E_1 , who directly receives the information from na_1 when a moving object does not behave accordingly to the normality constraints of c_{1y}^1 .

In E_2 , besides deploying the normality agent na_2 that manages the knowledge base c_{1y}^2 used to monitor trajectories, na_3 has been instantiated in order to monitor the speed of moving objects (c_{2y}^2).

The knowledge inferred by these agents is used by ena_2 to study the normality of every moving object detected in E_2 as the union of the normality analyzed by na_2 and na_3 . Regarding E_3 , the normality analysis is the same than in E_2 and the normality agents na_4 and na_5 and the environmental normality analysis agent ena_3 are responsible for that.

At this point, the agents ena_1 , ena_2 , and ena_3 manage the knowledge related to the normality analysis of E_1 , E_2 , and E_3 , respectively, which is send to the global normality analysis module GNAM. The interface of this module is represented by the fusion manager agent fma . Within this context, two fusion agents fa_1 and fa_2 have been instantiated to deal with the normality of c_1 and c_2 from a global point of view. The global information of vehicles and pedestrians provided by the global translator agent ta , enabling the global identification to, for instance, a pedestrian who crosses the pedestrian crossing simultaneously monitored in E_2 and E_3 .

Finally, the global fusion agent ga is instantiated in order to gather the knowledge obtained by the fusion agents fa_1 and fa_2 . In this particular surveillance system, ga only gets the conclusions of these agents and notify them to the decision support and crisis manager agent $dssa$. Currently, $dssa$ activates visual alarms in the monitoring tool subscribed to such channel (see Fig. 9).

6.3. Experimental validation

This section shows the experimental results obtained after having implemented and deployed the developed architecture prototype in the sub-environment E_2 of the scenario described in Section 6.1. The reader is encouraged to visit the prototype website² to watch a video demo of the artificial system. In this video, the reader can appreciate how certain abnormal situations are detected, such as pedestrians who cross the street but do not use the pedestrian crossing. Another kind of abnormal situations detected involve vehicles driving too fast along the roundabout. An extensive qualitative analysis of the analysis of trajectories and speed is discussed in depth in Albusac et al. (2009a,b), respectively.

The goal of the conducted experiments is to empirically show how the developed prototype supports the desired systematic requirements discussed in Section 4.3. Particularly, both the scalability and robustness of the system have been considered when deploying the prototype to monitor a real environment. The processing and communication times spent by the agents deployed in a single sub-environment have been measured in each of the conducted experiments.

In the case of processing times, although they directly depend on the internal implementation of the agents, we evaluate how the

² <http://oreto.esi.uclm.es/aplicaciones/oculus>

Fig. 9. Particular architecture of the surveillance system deployed to monitor the environment of Fig. 7.

workload is distributed when using multiple replicas of the same agent. This is carried out to check how scalability and robustness improve. In the case of communication times, we evaluate how the architecture scales when the number of messages exchanged between the agents increases.

Next, the employed technologies are summarized:

- The middleware ZeroC Ice (Henning, 2004) has been used to support all the architecture infrastructure.
- The relational database engine MySQL manages the persistent state of the surveillance agents.
- Java has been chosen to implement pa_2 , na_2 and na_3 .
- The script language Python was used to implement the blackboard system ena_2 .
- Finally, the management agents sm , pm and cm were implemented in C++.

In the current prototype, pa_2 , na_2 , na_3 , and ena_2 have been replicated with a dual goal: (i) to ensure the system robustness against failures and (ii) to distribute the workload of the surveillance process. Fig. 10 depicts the deployed network composed of three heterogeneous computers where the nodes *Oreto1*, *Oreto2* and *Oreto3* run. Firstly, *Oreto1* hosts the blackboard, the service used to deploy the event channels and the statistics agent. Secondly, *Oreto2* hosts two replicas of pa_2 , two replicas of na_2 , two more of na_3 , and ena_2 . This node runs a higher number of replicas because it is the most powerful computer within the

network. Finally, *Oreto3* hosts single replicas of pa_2 , na_2 , na_3 , and ena_2 . The mechanism used to maintain the state consistency between multiple replicas of a single agent is based on a master–slave scheme.

The set of experiments has been conducted under the following assumptions:

- The monitored scene lasts 2 min and 33 s, recorded at a frame rate of 25 frames per second. Approximately, 2500 of the 2830 frames contain moving objects.
- The frame rate can take the values of 10, 15, 20, or 25.
- Processing and communication times have been measured for each test.
- Three sets of tests have been conducted: (i) deploying a single node (*Oreto1*) with no replication, (ii) deploying a single node (*Oreto1*) with replicated agents and (iii) deploying the architecture of Fig. 10 with replicated agents.

Table 2 shows the times measured in seconds for 12 experiments. Tests 1–4 represent the first set, tests 5–8 represent the second test and tests 9–12 involve the third set.

Fig. 11 graphically shows the time spent in each task with no replication (tests 1, 3 and 4 of Table 2). The most significant portion of time is represented by the computation time of the preprocessing agent. This fact is mainly due to that such an agent calculates, for each detected situation, the area/s where every single moving object is located and classifies them. In the developed prototype,

Fig. 10. Designed network to deploy and evaluate the architecture prototype when monitoring the environment of Fig. 7.

Table 2

Set of conducted experiments varying the rate of fps and the nodes location (L=localhost, LR=localhost with replication, D=distributed).

Test	T	Ev/s	PA _{TP}	TA _{TC}	TA _{TP}	SA _{TC}	SA _{TP}	B_TA _{TC}	B_SA _{TC}	B _{TP}	ENA _{TC}	ENA _{TP}	T _{total}
1	L	10	242.12	4.09	1.53	4.07	0.92	5.27	4.87	7.33	6.89	3.41	350
2	L	15	234.62	4.84	1.47	4.93	0.95	5.22	4.86	7.54	6.90	3.16	223
3	L	20	234.04	5.72	1.34	5.70	0.88	5.16	5.06	8.21	6.82	3.20	176
4	L	25	253.56	7.29	1.22	7.11	0.73	5.25	5.08	9.36	6.77	3.22	140
5	LR	10	340.90	5.80	1.34	5.83	1.50	5.16	5.54	8.04	7.38	3.89	350
6	LR	15	371.67	8.05	1.88	6.71	1.69	5.89	5.72	9.69	8.13	4.05	224
7	LR	20	394.70	13.10	1.79	12.35	3.05	6.44	6.26	14.81	8.37	4.21	176
8	LR	25	427.88	15.86	2.65	16.81	2.98	5.87	6.16	19.01	8.33	3.92	141
9	D	10	209.19	16.751	5.939	14.449	5.817	21.216	18.616	27.764	23.623	13.557	351
10	D	15	192.994	16.429	6.521	13.632	6.177	21.525	18.842	28.643	24.975	14.199	224
11	D	20	185.819	16.053	6.62	14.154	6.786	16.146	15.408	26.217	17.775	13.657	255
12	D	25	189.011	20.903	6.731	18.284	6.293	17.994	17.520	32.082	22.730	13.484	140

Fig. 11. Time spent (in seconds) by each one of the surveillance agents with no replication in carrying out their respective tasks.

this first task is specially expensive because all the pixels of the object's lower hall are computed. On the contrary, the processing and communication times of normality agents and blackboards are short. Fig. 11 also shows that the architecture scales well when the event rate is increased since the time proportions remain constant.

On the other hand, Fig. 12 graphically shows the measured times when replicating agents according to the network architecture of Fig. 10. From a general point of view, three main differences are distinguished: (i) reduction in preprocessing times, (ii) increase in normality analysis times and (iii) increase in communication times.

Fig. 12. Time spent (in seconds) by each one of the surveillance agents maintaining a replication scheme in carrying out their respective tasks.

The first difference is justified by means of the adopted replication scheme that allows to distribute the workload in a transparent way. The second one is due to the use of a common database for the normality agents to store the monitoring results in every moment. This potential bottleneck can be attacked by using a distributed database. Finally, the communication times are increased since there are more replicas, involving a higher number of remote invocations to synchronize their state. Furthermore, the blackboard system implements a mutual exclusion algorithm to avoid the simultaneous access on behalf of normality agents. This fact increases the communication times with the blackboards, specially when deploying nodes in different computers.

7. Conclusions and future work

This work has discussed the multi-agent architecture proposed to support the design and development of intelligent surveillance systems based on a scalable normality analysis model (Albusac et al., 2009a). The agents that compose the architecture have been conceived to cover the needs posed by the model in which it is inspired and the communication mechanisms needed for the agents to carry out a global surveillance of the environment have been defined.

The implemented prototype of the proposed architecture has been used to deploy an intelligent surveillance system to monitor a real urban traffic environment in order to analyze the trajectories and speed of moving objects. This environment has been structured into three different sub-environments to simplify the monitoring process and to justify how the architecture give support to the agents and communication mechanisms needed to carry out the global surveillance.

In comparison to other solutions previously discussed in Section 2.3, which are also multi-agent based, the proposed architecture represents a flexible approach since one of the key concepts is the normality analysis of different concepts or events of interest by means of the normality agents. In this way, a surveillance system deployed by using this architecture can be easily extended when a new aspect must be monitored.

The set of conducted tests demonstrate how the architecture meets relevant surveillance requirements, such as scalability and robustness. This is particularly important when dealing with high

demand computation processes (e.g. real-time tracking) or when new events of interest must be studied. The implemented prototype also performs well when the rate of analyzed situations per second is increased and the monitored environment requires a more precised analysis, such as in indoor environments.

The systematic requirements that any surveillance system should support are considered by the proposed architecture and implemented by means of the developed prototype. However, some of them must be further studied. For instance, surveillance sensors other than security cameras must be included so that integration can be evaluated adequately. Within this context, we are deploying a surveillance system in a simulated health-care environment by using cameras, presence sensors and RFID readers.

Currently, we are interested in incorporating negotiation and argumentation abilities to the normality agents as an alternative to the information fusion provided by the fusion agents. Our hypothesis is that the normality agents try to convince to establish who among them is reasoning in a more precise way when monitoring the environment.

Acknowledgements

This work has been funded by the Regional Government of Castilla-La Mancha (Research Project PII1C09-0137-6488) and by the Spanish Council for Scientific Research (Research Project TIN2009-14538-C02-02).

References

- Abreu, B., Botelho, L., Cavallaro, A., Douchamps, D., Ebrahimi, T., Figueiredo, P., 2000. Video-based multi-agent traffic surveillance system. In: Proceedings of the 2000 IEEE Intelligent Vehicles Symposium, pp. 457–462.
- Albusac, J., Vallejo, D., Jimenez-Linares, L., Castro-Schez, J.J., Rodriguez-Benitez, L., 2009a. Intelligent surveillance based on normality analysis to detect abnormal behaviors. *International Journal of Pattern Recognition and Artificial Intelligence* 23 (7), 1223–1244 (Special issue on Visual Analysis and Understanding for Surveillance Applications).
- Albusac, J., Castro-Schez, J.J., Lopez-Lopez, L.M., Vallejo, D., Jimenez, L., 2009b. A supervised learning approach to automate the acquisition of knowledge in surveillance systems. *Signal Processing* 89 (12), 2400–2414 (Special issue on Visual Information Analysis for Security).
- Bovenkamp, E.G.P., Dijkstra, J., Bosch, J.G., Reiber, J.H.C., 2004. Multi-agent segmentation of IVUS images. *Pattern Recognition* 37 (4), 647–663.

- Castanedo, F., Garcia, J., Patricio, M.A., Molina, J.M., 2008. A multi-agent architecture to support active fusion in a visual sensor network. In: Second ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2008), pp. 1–8.
- Chen, B., Cheng, H.H., Palen, J., 2009. Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. *Transportation Research Part C* 17 (1), 1–10.
- Collins, R., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., 2000. A system for video surveillance and monitoring: VSAM Final Report. Robotics Institute, Carnegie Mellon University.
- Detmold, H., van den Hengel, A., Dick, A., Falkner, K., Munro, D.S., Morrison, R., 2008. Middleware for distributed video surveillance. *IEEE Distributed Systems Online* 9 (2), 1–11.
- Enficiaud, R., Lienard, B., Allezard, N., 2006. Clovis-a generic framework for general purpose visual surveillance applications. In: *IEEE Workshop on Visual Surveillance*, pp. 177–184.
- Engelmore, R., Morgan, A., 1988. *Blackboard Systems*. Addison-Wesley.
- Forsyth, D.A., Ponce, J., 2002. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference.
- Foundation for Intelligent Physical Agents, FIPA, 2004. *FIPA Agent Management Specification*, <<http://www.fipa.org/specs/fipa00023/index.html>>.
- Garcia, J., Carbo, J., Molina, J.M., 2005. Agent-based coordination of cameras. *International Journal of Computer Science and Applications* 2 (1), 33–37.
- Haritaoglu, I., Harwood, D., Davis, L.S., 2000. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8), 809–830.
- Henning, M., 2004. A new approach to object-oriented middleware. *IEEE Internet Computing* 8 (1), 66–75.
- Hu, W., Xie, D., Tan, T., 2004. A hierarchical self-organizing approach for learning the patterns of motion trajectories. *IEEE Transactions on Neural Networks* 15 (1), 135–144.
- Karlsson, B., Backstrom, O., Kulesza, W., Axelsson, L., 2005. Intelligent sensor networks – an agent-oriented approach. In: *Proceedings of the Workshop on Real World Wireless Sensor Networks*.
- Lee, R.S.T., 2003. iJADE surveillant – an intelligent multi-resolution composite neuro-oscillatory agent-based surveillance system. *Pattern Recognition* 36 (6), 1425–1444.
- Patricio, M.A., 2007. Multi-agent framework in visual sensor networks. *EURASIP Journal on Advances in Signal Processing* 2007, 1–21.
- Remagnino, P., Shihab, A.I., Jones, G.A., 2004. Distributed intelligence for multi-camera visual surveillance. *Pattern Recognition* 37 (4), 675–689.
- Saykol, E., Gudukbay, U., Ulusoy, O., 2010. Scenario-based query processing for video-surveillance archives. *Engineering Applications of Artificial Intelligence* 23 (3), 331–345.
- Smith, G.J.D., 2004. Behind the screens: examining constructions of deviance and informal practices among CCTV control room operators in the UK. *Surveillance and Society* 2 (2/3), 376–395.
- Stringa, E., Regazzoni, C.S., 2000. Real-time video-shot detection for scene surveillance applications. *IEEE Transactions on Image Processing* 9 (1), 69–79.
- Tao, D., Li, X., Wu, X., Maybank, S., 2007. General tensor discriminant analysis and Gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (10), 1700–1715.
- Tian, Y., Brown, L., Hampapur, A., Lu, M., Senior, A., Shu, C., 2008. IBM smart surveillance system (S3): event based video surveillance system with an open and extensible framework. *Machine Vision and Applications* 19 (5), 315–327.
- Valera, M., Velastin, S.A., 2005. Intelligent distributed surveillance systems: a review. *IEEE Proceedings-Vision, Image and Signal Processing* 152 (2), 192–204.
- Vallejo, D., Albusac, J., Jimenez, L., Gonzalez, C., Moreno, J., 2009. A cognitive surveillance system for detecting incorrect traffic behaviors. *Expert Systems with Applications* 36 (7), 10503–10511.
- Velastin, S.A., Remagnino, P., 2005. Intelligent distributed video surveillance systems. *The Institution of Electrical Engineers (IEEE)*.
- Velastin, S.A., Boghossian, B.A., Lo, B.P.L., Sun, J., Vicencio-Silva, M.A., 2005. Prismatic: toward ambient intelligence in public transport environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A* 35 (1), 164–182.
- Wooldridge, M., Jennings, N.R., 1995. *Intelligent Agents: Theory and Practice*. Knowledge Engineering Review 10 (2), 115–152.
- Xu, D., Yan, S., Tao, D., Zhang, L., Li, X., Zhang, H.J., 2006. Human gait recognition with matrix representation. *IEEE Transactions on Circuits and Systems for Video Technology* 16 (7), 896–903.
- Yager, R.R., 1988. On ordered weighted averaging aggregation operators in multi criteria decision-making. *IEEE Transactions on Systems, Man and Cybernetics* 18 (1), 183–190.