Research Article

Intelligent Surveillance for Understanding Events in Urban Traffic Environments

D. Vallejo, F. J. Villanueva, J. A. Albusac, C. Glez-Morcillo, and J. J. Castro-Schez

Faculty of Computing, University of Castilla-La Mancha, 13071 Ciudad Real, Spain

Correspondence should be addressed to D. Vallejo; david.vallejo@uclm.es

Received 27 January 2014; Revised 3 May 2014; Accepted 30 July 2014; Published 14 August 2014

Academic Editor: Jesus Martinez del Rincon

The main goal of intelligent surveillance consists in providing security systems with the skills required for correctly detecting and analyzing events in monitored environments. Since these environments are complex and the information is distributed through them, the use of agent-based approaches has become more and more popular when monitoring moving objects. This paper describes how an existing agent platform has been adopted and used to give support to intelligent surveillance systems. Agents deployed by means of this platform implement a behavior-based model that is flexible enough to deal with the challenges that monitored environments pose. Two case studies of urban traffic environments are discussed to prove the feasibility of the proposal.

1. Introduction

The evolution of artificial intelligence (AI) techniques has made possible the design and development of intelligent surveillance systems [1] that are able to monitor environment from a high-level of abstraction; that is, systems that are able to detect and understand real-time complex events. The two key aims to reach by means of the use of these methods are the real-time automatic analysis of scenes and the obtaining of detailed and precise information that allows the artificial system to make decisions and anticipate future events. Intelligent surveillance systems often rely on techniques based on image processing and visual reasoning, which can reduce the human resources cost and solve the mentioned problems of tiredness and fatigue by supporting security staff. Depending on the abstraction level of surveillance systems, there currently exist systems capable of detecting crowds [2], identifying suspicious or lost items [3], inferring anomalous behavior [4], or even analyzing object trajectories [5] from visual information. These systems have experienced a great evolution during the last few years but there is still a lot of work to do.

Recently, research on this field has been mainly increased due to the huge society demand for this kind of systems both in public and private spaces. One of the most representative examples is the city of London, where the number of surveillance cameras per citizen is very high due to the likelihood of terrorist attacks.

From the architectural point of view, intelligent surveillance systems are often deployed on top of a multilayer architecture, where each layer performs a well-defined function and generates a set of results that is used as input for the rest of layers (see Figure 1). In this kind of architectures, AI techniques are generally applied to model, formalize, develop, and implement artificial systems capable of supporting the analysis of situations, learning from data, and making decisions consequently.

Usually, researchers focus their work on a specific stage or layer, but most work is carried out on low-level stages such as segmentation and tracking, that is, the image processing mechanisms used to partition a digital image into multiple regions and to locate moving objects in a scene, respectively. In higher levels, research is commonly oriented to solve particular problems, such as detecting anomalous behavior or identifying suspicious or lost objects. These approximations serve as the starting point for developing intelligent systems applied to concrete problems, but they lack scalability and flexibility to apply them in more general surveillance
environments, which require the analysis of a wide variety of objects, concepts, or elements. In fact, most of these systems are prototypes that are not industrially exploited as the consequence of the number of wrong alarms generated due to incorrect interpretations. For instance, an urban traffic scenario may be studied to monitor vehicle speed or analyze if drivers behave in a suspicious way. Within this context, the design of an intelligent surveillance system that allows us to dynamically embed modules when needed could be desirable to face the monitoring of different issues within a common domain or environment.

These high-level layers are responsible for correctly detecting and understanding and are the main contributor when advancing the state of the art of intelligent surveillance systems. Thus, previous experience and knowledge to monitor environments are required. This knowledge can generally be given by human experts in the security domain or learned from past situations.

In order to acquire the expert knowledge and to learn from previous experience, knowledge acquisition tools and machine learning techniques can be respectively used. Knowledge acquisition tools require a human expert to explicitly define the domain knowledge. In the context of surveillance, this knowledge may involve the visual definitions of the physical zones that compose the monitored environment or the main characteristics of moving objects, such as size, shape, or color. On the other hand, machine learning techniques make use of automatic or semiautomatic algorithms to infer the domain knowledge needed to carry out the reasoning.

Another of the key factors to have success with intelligent surveillance is the correct management of uncertainty and vagueness. In other words, the model used to represent the environment and the surveillance tasks must be able to support these two concepts with the aim of providing a coherent solution to the real world. A comparison can be established with a security guard who perceives a blurred environment and cannot ensure what happens with an absolute certainty.

Last generation surveillance systems [1] are also characterized by the huge number of security devices distributed all around the environment. In fact, these devices can belong to different classes, that is, video cameras, microphones to record ambient sounds, and other sensors, such as presence detectors or magnetic readers, and they can coexist within a common scenario. Consequently, the information obtained from the different surveillance sources is distributed so that we need to use a scheme that facilitates its fusion in order to provide an advanced surveillance (see Figure 2).

To address this problem and give support to the intelligent surveillance modules, we make use of a multiagent architecture to deploy intelligent agents specialized in the different surveillance services required by the environment to be monitored. The use of a multiagent architecture makes easy the design of a solution based on the intelligent management of distributed knowledge by offering a scalable and flexible system when integrating new concepts or elements that allow us to provide a more sophisticated surveillance. Thus, the major contribution of this paper lies in the monitoring of complex urban environments through an approach based on intelligent agents that use knowledge bases that define how to understand situations. Plus, these agents are supported by a platform that scales when the scene complexity grows, which is essential in surveillance domains.

The rest of the paper, which extends the work discussed in [6], is structured as follows. Section 2 carries out a study of intelligent surveillance systems and overviews some relevant proposals. In Section 3, the agent-based model used to monitor surveillance environments is discussed. Next, Section 4 discusses several case studies which discuss how a multiagent system was deployed to monitor urban traffic environments where pedestrians and vehicles continuously move. Finally, Section 5 concludes the paper and proposes several future research lines.

2. Related Work

Intelligent surveillance can be understood as processing information from the environment in order to monitor the events and situations that take place. This task has been traditionally assigned to the security personnel. The typical example is intelligent video surveillance, where the input is one or more video streams from cameras and the output aims at solving a concrete problem, such as crowd detection, face recognition, or even behavior analysis (for a complete survey, see [1, 5]), which involves visual reasoning. The remainder of this section shows an overview of relevant intelligent surveillance systems from two points of view: behavior analysis and agent-based approaches.
2.1. Visual Surveillance and Behavior Analysis. One of the most significant research lines within the context of intelligent surveillance is behavior analysis, which is often focused on detecting anomalous events. Usually, research in this field has been almost exclusively concerned with the analysis of individual domains (vehicle traffic, parking, halls, etc.) or aspects (paths, actions, movements, etc.). Although this research line offers effective and efficient solutions for concrete problems, it lacks a global view of the environment where multiple aspects must be taken into account. For example, consider that the behavior of a vehicle depends on trajectory analysis and speed analysis and both aspects must be considered at the same time to analyze the global behavior of the vehicle.

There exists a wide range of methods and techniques used for behavior analysis and understanding. In this conceptual layer there is no general consent to analyze behaviors as the huge number of approaches used shows. However, authors in this field choose people and vehicles to represent and understand behaviors. Another common issue is the analysis of paths followed by moving objects, which can be directly translated into the surveillance domain. The reasoning mechanisms are generally based on processing the visual information obtained from video cameras deployed around the monitoring environment. One of these techniques is Dynamic Time Warping (DWT), which allows us to measure the similarity between two sequences that differ in time or speed [7].

Another approach consists in adopting a finite state machine (FSM), which is composed of states, actions, and transitions. Within the context of intelligent surveillance, the states represent the object situation on the environment, the actions represent moving object events, and the transitions refer to actions made by moving objects that have been recognized by the artificial system. Some applications of this technique involve the control of a robot monitored by a camera [8] or vehicle behavior understanding from aerial cameras [9]. On the other hand, the use of context-free grammars [10] has been also applied to intelligent surveillance in order to generate context-free languages used to recognize behaviors in a monitored environment [11].

Currently, one of the most widespread methods to deal with behavior analysis is Hidden Markov Models (HMM) [12]. Essentially, a HMM consists in a statistical model composed of a set of hidden states, the observable data, the transition probability, and the output probability. The goal of a HMM is determining the hidden parameters depending on the observable data. The extracted model parameters can be used to perform further analysis such as the case of pattern recognition applications. Research on video surveillance integrates the HMM to build artificial systems that help to automate the task of behavior analysis [13, 14].

There exists a significant number of research works focused on behavior analysis, such as discussed in [15, 16]. Unfortunately, these works do not define scalable mechanisms for dealing with surveillance depending on different concepts, such as speed analysis, path analysis, or activity analysis. In other words, they are often focused on particular aspects. In contrast to this approach, this work solves these limitations by means of normality concepts [17].

A normality concept refers to the definition of a surveillance task for a concrete aspect and its instantiation for a particular environment. For instance, if a surveillance system needs to analyze speed, then the speed concept definition can be embedded into the surveillance system to carry out the speed normality analysis. In the same way, if the artificial...
system needs to analyze suspicious or forgotten objects in a monitored environment (e.g., in an airport), then the system will be provided with the definition of the normality concept that addresses this issue. Besides, all these local normality analyses affect the global normality analysis of the environment.

2.2. Agent-Based Approaches. Since intelligent surveillance may involve a large number of moving objects whose behavior must be analyzed from the data obtained from multiple sensors, a multiagent based approach fits very well to deal with the design of an intelligent surveillance system. Thus, agents give support to the distributed services that compose the whole system. One of the first approximations in the area of intelligent surveillance through multiagent systems was to combine the information obtained from multiple cameras. An interesting work was introduced in [15], where the authors proposed a multiagent architecture to get information of scenes from different points of view.

One of the main goals was to provide the final users with services that can be customized depending on several parameters, such as the number of cameras, the orientation, or the video events to classify behaviors. However, and although the system aimed at supporting different video services, there were not well-defined services to discover, manage, or compose services. To address these issues, the notion of explicit surveillance services is proposed in [18]. A similar work was discussed in [19], in which a layered architecture is proposed. Low-level layers consist of sensors and actuators and each camera is represented by a proxy agent, which manages the video events. On top of these layers, there exist different high-level agents that provide image processing services.

In [20] a more recent work is discussed, which is mostly focused on using the multiagent technology for coordinating the tracking of moving objects. This coordination is based on the exchange of high-level messages among agents that use a symbolic model to interpret situations.

Distributed intelligent surveillance can be faced from different points of view. For example, a researcher may use a general-purpose knowledge-based system to deploy concrete instances of that system for particular problems, such as a surveillance system. This is the approach followed in [21]. The paper presented an open and flexible architecture for a distributed knowledge-based system that can be applied to different problems. The architecture is based on multiagent technology and the use of ontologies to represent knowledge. To evaluate the system, the authors deployed a system for psychological disorders consulting and discussed the support ontology.

Current architectures for intelligent surveillance do not provide all the mechanisms needed to give support to surveillance components. For instance, they do not allow us to define, include, and instantiate surveillance components that analyze multiple aspects and their later integration by means of fusion information components. Although some of these architectures include machine learning algorithms, they do not perform an integration of such algorithms within the whole architecture, that is, regarding previously deployed components. Besides, they lack a well-defined mechanism for associating surveillance sensors and surveillance components. In relation to visual information, the image processing requires high computational load to be performed in real-time and, therefore, distributing these tasks between computational nodes is needed. This is our motivation for designing an integral architecture based on agent technology in order to overcome these limitations.

3. Understanding Events through Intelligent Agents

The agent model used in this work to monitor and understand events in urban traffic environments relies on the agent platform proposed in [22] and extends the work discussed in [6]. Basically, the agents of this architecture implement a behavior-based model so that the behaviors represent tasks to be carried out by the agents. From a general point of view, two kinds of behaviors can be distinguished:

(i) simple, which cannot be divided into other behaviors. Similarly, simple behaviors can be classified as follows:
   (a) cyclic, which are continuously executed until the agent is destroyed.
   (b) one shot, which are executed one single time;

(ii) composite, which are composed of other behaviors. Similarly, simple behaviors can be classified as follows:
   (a) sequential, so that all their subbehaviors are sequentially executed.
   (b) parallel, so that all their subbehaviors are concurrently executed.

The simplest behaviors are usually implemented as one shot or cyclic. Conversely, if a behavior is required to be structured into a number of sequential subbehaviors, then the correct choice would be a sequential behavior. Finally, parallel behaviors address the idea of concurrency to take advantage of multithreaded processors. Multiple behaviors can be considered at the same time when using, for example, parallel behaviors. From the monitoring point of view, a single agent could be monitoring several aspects or events of interest through multiple parallel subbehaviors. Each parallel subbehavior would run on a different physical thread.

Figure 3 depicts the control flow of our agents according to the behaviors that govern how they detect and analyzed events. Basically, the agents maintain a pool of behaviors that are sequentially executed. If the pool is empty, then the agent becomes idle. On the other hand, if the agent is required to execute a new behavior, then this is queued up for executing. This approach is quite flexible and allows a large number of configuration settings. If a behavior is not done due to an unexpected exception or a temporal constraint violation, for
example, then the behavior is automatically removed from the pool.

In the surveillance domain behaviors can be understood as the computational unit that comprises the logic or code responsible for monitoring an aspect or event of interest. For example, a behavior implemented by an agent A may be responsible for analyzing moving object trajectories, while another behavior implemented by an agent B may be responsible for analyzing moving object speed. These behaviors are defined from the point of view of a normality analysis [17]. The key idea is the normality concept, which comprises a number of constraints that must be satisfied by a monitoring object so that an agent can classify its behavior as normal. In other words, behaviors know how moving objects ideally behave according to normal events and if they do something different to these expected events, then the behaviors detect the anomaly. The current implementation of normality concepts relies on fuzzy logic to address the uncertainty of real environments.

On the other hand, the definition of normality is generally straightforward and the number of normal situations that can take place in a scenario is usually low and they are well known. Conversely, the number of anomalous situations that can happen is hardly predictable. That is why we chose modeling normality instead of abnormality.

Independently of the type of behavior that the agents implement, they typically will manage a knowledge base that will be used to understand events and actions. In some of our previous works, these knowledge bases were acquired thanks to the help of a human expert [17] or learned by means of machine learning algorithms [23]. Now, the agents make use of these knowledge bases to monitor the environment but the flexibility gets increased thanks to this wide range of behaviors.

Within the context of urban traffic environments, the number of moving objects that should be monitored is usually large. The reader is encouraged to consider a crowded street with multiple pedestrian crossings to get an idea of a potential monitored scene.

From the point of view of monitoring, we chose to deploy a single software agent when a new moving object appears in the scene. Thus, each agent is responsible for a single moving object, using its knowledge base to determine whether its behavior is correct or not. When the object leaves the scene, then the associated software agent is destroyed and the used resources are freed. An agent is aware of this situation when no events associated to its related moving object are detected during a configurable period.

Usually, agents that monitor environments make use of cyclic behaviors since they are continuously analyzing events. This is the approach adopted by the multiagent systems deployed in the case studies discussed in the next section. However, it is also possible to replace these behaviors by parallel behaviors, increasing the system performance and scaling according to the number of monitored objects.

Before addressing the experimental results obtained by means of the deployment of the system in two different scenarios, it is important to stress the benefits of the agent platform used in this work from an architectural point of view. To do that, a justification of how the architecture provides a set of relevant systematic requirements is included within the context of intelligent surveillance. The point of reference is the work discussed in [24], where the authors proposed a middleware for video surveillance and stated a well-defined set of systematic requirements. These requirements are as follows:

(i) scalability to facilitate the integration of new components,
(ii) availability to guarantee the system robustness and detect failures,
(iii) evolvability to respond to hardware or software changes,
(iv) integration to integrate new devices,
(v) security to fight against inadequate or unauthorized uses,

![Figure 3: Control flow of the agents, which relies on a pool of behaviors that are executed.](image-url)
Figure 4: (a, b, and c) Images captured by a surveillance camera in a crowded road. (d, e, and f) images analyzed by the agent-based surveillance system. The blue squares represent surveillance areas; the green points are associated with monitored objects. The text shows the area percentage covered by each object.

(vi) manageability to carry out 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 cameras and, on the other hand, the need of monitoring new events of interest or threads. This can be achieved by deploying new software agents where their knowledge base turns around new monitoring aspects.

Availability has been managed by means of replication mechanisms. In this way, if a nonreplicated agent that analyzes a concept or event of interest terminates abruptly, then the software components that uses its information will keep working but without the knowledge previously inferred by such an agent.

Regarding the security mechanisms provided by the platform, multiple methods have been considered protecting the information, ensuring its integrity, or verifying the identity of the agents. 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.

4. Experimental Results

This section discusses several environments where the proposed multiagent architecture for monitoring has been deployed.

The first case study is related to a specific urban traffic environment where a multiagent system was deployed to monitor moving objects. The monitoring process took place in an important route of a Hungarian city where two roads are separated by a sidewalk that allows the pedestrian to get on the tram. Essentially, our goal was to determine if every moving object is on the physical area of the environment where it was supposed to be at all times. For example, pedestrians are not supposed to invade the road unless they use the pedestrian crossings. On the contrary, vehicles must not invade the sidewalks in no way whatsoever.

Figure 4 depicts some frames taken from a surveillance camera deployed in aforementioned scenario. Note that the regions of interest were chosen taking into account where the monitored objects, vehicles, and pedestrians should ideally move. In particular, the closest areas to the camera were selected in order to get better results from the tracking process. The second row of images shows visual information generated by the software agents: blue squares that delimit the surveillance areas and green points that represent the monitored moving objects (the full video is available at http://www.esi.uclm.es/www/dvallejo/wi-iat). For each monitored object, the monitoring system shows the area percentage covered regarding the monitoring areas denoted by blue squares. This value is used for the agents to know where the moving objects are and whether they should be there or not.

In this particular case, the knowledge used by the software agents comprises the monitored areas, which are defined by using a graphical tool (see Figures 5 and 6) that simply
The knowledge acquisition tool depicted in Figure 5 can be used to obtain information about the vehicle trajectories.

Right: State of the virtual security guard in real-time. Left: Visual information obtained by means of the video cameras deployed in the environment. Center: Camera selected by the user of the application.

Figure 6: Surveillance tool developed to monitor the environment in real-time. Left: Visual information obtained by means of the video cameras deployed in the environment. Center: Camera selected by the user of the application. Right: State of the virtual security guard in real-time.

Figure 5: Knowledge acquisition tool developed to model normality aspects or events of interest.

Figure 6: Surveillance tool developed to monitor the environment in real-time.

The scenario and the moving objects is not high and, therefore, the quality of the results is not high as previously discussed. In this case study, the time period of the monitoring process took place at noon in a cloudy day. However, the system has been designed so that the computer vision subsystem can be replaced by a more sophisticated one if it is to be deployed on a real environment under more stressed situations and contrary weather conditions.

The second case study is related to a scenario that consists in a roundabout, in which entering traffic must yield to traffic already in the circulatory roadway (see the first row of Figure 7). The application of the artificial system in a concrete problem aims at justifying the use of surveillance models and architectures as the mechanism for automating and supporting the typical tasks done in a security control center, such as the analysis of vehicles behavior in the urban traffic domain. In this empirical case, the normality study will be made by a software agent that analyzes the concept of normal trajectories, that is, whether drivers behave correctly according to traffic law. In this case, we deploy a single software agent per aspect or event of interest to be monitored.

Figure 7 shows the temporal evolution of the reasoning process carried out by the software agent and establishes a comparison regarding the security guard’s reasoning in different moments. Next, this comparison will be described by paying special attention to how the agent emulates this reasoning model.

Before deploying the multiagent system and carrying out the normality analysis, the security expert user makes use of the knowledge acquisition tool to define the knowledge of the monitored environment. This knowledge implies the explicit definition of the regions that compose the urban scenario and the normal trajectories inside it. To do that, the expert identifies the allowed trajectories according to the traffic law and specifies the set of constraints that allows us to model these trajectories.

The normality analysis of the surveillance agent in the moment $t = 1$ is represented by the first column of Figure 7. As shown in the figure, two vehicles come to the roundabout. In this case, we will focus on the description of the normality analysis for the second moving vehicle in the moment $t = 1$, that is, regarding the black sedan that drives near the parked vehicles. Taking into account this situation, a hypothetical security guard would observe the scenario and would infer that the vehicle behaves correctly since it is driving on the right and coming to the roundabout so that the vehicle could be making one of the possible trajectories in that moment. These conclusions can be graphically appreciated in the third and fourth rows (see Figure 7). In the third row, the possible trajectories that the monitored vehicle (marked with a red ellipse) could be doing are shown: turn right on the roundabout, turn left on the roundabout, or turn around (each one of the green arrows represents a correct trajectory). On the other hand, the fourth row of the first column reflects the application of the normality model with regard to the concept of normal trajectory attached to the monitored vehicle. In this case, the analyzed concept is normal trajectories, denoted as $c_1$, and the instances of such concept currently associated to the vehicle, that is,
Figure 7: Normality analysis carried out by the multiagent surveillance system. The columns of the mosaic represent the temporal evolution of such analysis. (a) static images obtained from the video camera used to get the visual information from the environment. (b) knowledge inferred by a hypothetical security guard in each moment. (c) possible trajectories of the monitored vehicle (the green arrows represent normal trajectories and the red ones involve forbidden trajectories). (d) situation of the normality analysis model depending on the trajectory concept previously defined by the human expert. (e) visual representation of the constraints associated to the possible trajectories of the monitored vehicle.
the instantiated trajectories for the environment, are $c_{11}$, $c_{12}$, and $c_{13}$.

Finally, the images of the fifth row represents in a visual way the satisfaction degree of the constraints associated to the trajectories that the vehicle could be carrying out. In this case, the floor region drawn in a green gradient represents the satisfaction degree of the spatial constraints, that is, whether the vehicle is over some of the physical regions that were defined for a normal trajectory or, on the contrary, is over a region that does not match a correct trajectory. In the first moment, the vehicle is over the beginning area of three possible trajectories, which end on the physical areas $a_{13}$ ($c_{13}$), $a_{2}$ ($c_{12}$), and $a_{4}$ ($c_{11}$). Dark green tones indicate the physical locations expected for the vehicle in a concrete moment while light green tones indicate physical locations far from the location expected for the vehicle. This information is used to visually appreciate the spatial constraints satisfaction. Thus, the length of the vectors $d_{13}$, $d_{2}$, and $d_{4}$ represents the distance regarding the end areas, $a_{13}$, $a_{2}$, and $a_{4}$, associated with the trajectories $c_{13}$, $c_{12}$, and $c_{11}$, respectively. This information serves to visualize the distance constraints for each of the trajectories. For instance, the vector $d_{4}$ is the larger one because the trajectory $c_{11}$ has the larger route, since it implies turning around and reaching the region $a_{4}$.

In the second time moment (second column in Figure 7), the vehicle has moved to the roundabout. If a security guard analyses this frame, he will infer that the vehicle keeps on driving in a normal way as it has reached the roundabout and it will turn, in theory, right, left, or around. In this way, the software agent keeps on associating the previous three trajectories with the monitored vehicle, since the normality constraints defined for each one of them are satisfied. In the last row of the second column, the green area over the entrance of the roundabout is visually appreciated as well as the length reduction of the distance vectors, since the distances regarding the end areas of the associated trajectories have also been decreased.

The third time instant reflects the more interesting moment from the normality analysis point of view in the traffic urban environment. The vehicle chose to turn left on the roundabout but it drives along the wrong way because it left the roundabout to the left instead of the right. In this moment, the security guard would have detected the driving offence because the vehicle did a manoeuvre which is not allowed by the traffic law (vehicles drive on the right in Spain as the general rule.). It is important to remark that the security guard cannot stop the monitor to detect this anomalous behavior, since this concrete driving offence is done in a short period of time ranging from five to ten seconds. This problem was previously introduced and it is one of the reasons that justifies the design of intelligent surveillance systems that deal with visual information.

As performed in previous time moments, the software agent checks the satisfaction degrees of the constraints of the previously associated trajectories. In this moment, the physical location of the vehicle violates the spatial constraints of all trajectories, since the vehicle location belongs to a region which is not included in the normal trajectories defined for this environment. Therefore, the agent detects that the vehicle has no normal trajectory associated and it is able to infer that the vehicle does not behave normally according the concept of normal trajectory or, in other words, the vehicle is carrying out an anomalous behavior. As can be shown in Figure 7, the conclusion obtained by means of the visual reasoning process equals the one obtained by the hypothetical security guard. In the last row of this column, it can be appreciated that the vehicle is not located over some of the three regions expected for the trajectories that were defined according to the normality of the urban traffic environment. Thus, the multiagent system will determine that the current situation is not normal.

The quality of the obtained results in this second case study are better than in the first one since the output data of the used OpenCV tracking plugin is clearer.

Finally, it is important to remark that the administration, configuration, and deployment of MAS developed by using the agent platform is straightforward thanks to the graphical tool integrated in ZeroC ICE [25], the used communication middleware, which is depicted in Figure 8. Basically, this tool facilitates the definition of servers, services, and nodes to automatically deploy a distributed system, such as the one discussed in this paper. A server provides a number of services (agents in this case) and are deployed on nodes, which are associated to the actual physical computers. Furthermore, this graphical tool can be also used to define replica

![Figure 8: Screenshot of the graphical user interface employed to configure and deploy MAS by means of the agent platform.](image-url)
groups, that is, entities that allow the replication of services when the computational complexity of tasks gets increased. For example, if the number of monitored areas or moving objects grows, then the replica groups guarantee that extra computational power can be used to address this kind of situations.

This approach makes it possible to distribute the computing load associated to the behaviors implemented by the agents across multiple servers. Thus, if the number of monitored moving objects or events of interest gets increased and, consequently, the number of agents, then new nodes can be deployed to address the system scalability.

5. Conclusions and Future Work

Due to the society needs for living in secured environments, intelligent surveillance systems are getting more and more important. Therefore, the investment on this research line is becoming increasingly relevant with the aim of developing more sophisticated surveillance systems and architectures. One of the more promising research lines consists in emulating the security guards’ behavior, that is, defining reasoning models that give support to the tasks done by the security staff of particular environments.

In this work we have discussed an agent-based approach that can easily be used in surveillance systems to increase the flexibility when monitoring aspects or events of interest. To carry out this task, the agents implement a number of behaviors that can adapted to the requirements of intelligent surveillance systems. These behaviors turn around knowledge bases, which can be defined by human experts or learned from previous experience. In order to increase the functional capabilities of intelligent surveillance systems, it is essential to understand what moving objects do in the monitored environments. If this can be done in an artificial way with a high accuracy, the limitations of human monitoring (e.g., fatigue and tiredness) can be overcome.

The multiagent architecture has been used to deploy surveillance systems on multiple urban traffic scenarios. To do that, the visual information provided by video cameras placed at such scenario has been processed. The description of the results justifies the importance of this kind of tools to support the staff responsible for the security of environments by solving the problems of fatigue and tiredness due to the continuous observation of surveillance monitors.

We are currently integrating previous work on intelligent surveillance [17, 23] into the agent architecture so that every surveillance task is a behavior executed by an agent. After that, our intention is focusing our work on parallel behaviors to reach all the power of multicore computing and increase the performance of our surveillance system.

On the other hand, one of the future research lines that is currently being addressed is the use of surveillance sensors of different nature within a common environment in order to complement the visual information provided by video cameras. This research line, mainly focused on indoor environments, will allow us to offer more sophisticated surveillance by fusing the information obtained from multiple surveillance sources.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the Spanish Ministry of Science and Innovation and CDTI through projects DREAMS (TEC2011-28666-C04-03), ENERGOS (CEN-20091048), and PROMETEO (CEN-20101010).

References


Submit your manuscripts at
http://www.hindawi.com