This document is published in:

Bajo Pérez, J. et al. (eds.), (2012). *Highlights on Practical Applications of Agents and Multi-Agent Systems*: 10th International Conference on Practical Applications of Agents and Multi-Agent Systems. Advances in Intelligent and Soft Computing: 156. Berlin, Germany: Springer-Verlag, pp. 103-112.

DOI: 10.1007/978-3-642-28762-6_13

Distributed Active-Camera Control Architecture Based on Multi-Agent Systems

Alvaro Luis Bustamante, José M. Molina, and Miguel A. Patricio

Abstract. In this contribution a Multi-Agent System architecture is proposed to deal with the management of spatially distributed heterogeneous nets of sensors, specially is described the problem of Pan-Tilt-Zoom or active cameras. The design of surveillance multi-sensor systems implies undertaking to solve two related problems: data fusion and coordinated sensor-task management. Generally, proposed architectures for the coordinated operation of multiple sensors are based on centralization of management decisions at the fusion center. However, the existence of intelligent sensors capable of taking decisions brings the possibility of conceiving alternative decentralized architectures. This problem could be approached by means of a Multi-Agent System (MAS). In specific, this paper proposes a MAS architecture for automatically control sensors in video surveillance environments.

1 Introduction

Nowadays, video surveillance systems are evolving towards complex information systems, being capable of providing the operator with a great amount of data obtained through spatially distributed sensor networks. The advances of the underlying technologies like digital communication, video coding and transmission, and specially, wireless sensors and actuator networks (WSANs) [1] have lead an easy deployment of new video surveillance sensors for its use in environmental monitoring, health-care systems, homeland security, public safety, and in general critical environments.

There is many scientific research done to take advantage of the visual sensor networks, and also commercial systems, to provide rich features in video surveillance like object detection and tracking, object and color classification, activity recognition, alert definition and detection, database event indexing, and so on [7].

Alvaro Luis Bustamante · José M. Molina · Miguel A. Patricio
Universidad Carlos III de Madrid, Avda. de la Universidad Carlos III,
22. 28270 Colmenarejo, Madrid
e-mail: {aluis,mpatrici}@inf.uc3m.es,molina@ia.uc3m.es

But considering an environment plenty of Pan-Tilt-Zoom (PTZ) cameras with some existing overlapped fields of views, it makes sense to apply some kind of coordination, competition, and collaboration among PTZ cameras to satisfy or improve some established goals. The benefits of active cooperative sensing as compared to non-cooperative sensing has been well established in literature [5]. There are some interesting dual-camera frameworks proposed in which a master camera takes wide panoramic images and the slave zooms into the targets to get more accurate images, like those described in [11, 5]. More recent work like in [14] describes a method for the dynamic target assignment according to the availability and accuracy of each camera. In [8] is addressed the generic problem of collaboratively controlling a limited number of PTZ cameras to capture an observed number of subjects in an optimal fashion. However, the problem of these approaches is the camera control centralization in a single node, leading the architectures to fail in scalability, performance and fault-tolerance.

The design of this kind of visual surveillance systems implies undertaking to solve two main problems [10]. The first one is the data fusion, which is related with the combination of data from different sources in an optimal way [9]. The second one is the multi-sensor management. It assumes that the previous problem was solved, and it is in charge of optimizing the global management of the joint system through the application of individual operations in each sensor [12].

There are two main approaches to solve this problem, the centralized and the decentralized way. A centralized architecture uses to be based on a data fusion center or node of the network combining the whole system information to planify and execute actions over each sensor. Thus, a centralized architecture is affordable to build a prototype but have many problems related with scalability, and deployment when the sensors are highly distributed [12].

This research is focused on solving the two main problems described before, multi-sensor management and data fusion for PTZ cameras in a distributed way, since we assume that sensors may have a high degree of autonomy, so that the last decisions about task to be executed in the sensor is taken in their own management system [15]. In this way, this problem could be approached by means of a Multi-Agent architecture, as proposed in this paper.

The study of multi-agent systems (MAS) focuses on systems in which many intelligent agents interact with each other. So, this work proposes a MAS architecture for support advanced controls for active camera sensors in the video surveillance environment. There are some previous works done by our research group also facing the problem of still/active cameras management via MAS, like the proposed in [6, 4, 13]. However this previous works do not contemplate the data fusion as an integral part of the architecture itself and its management with MAS. This was presented in [3] but for different purposes, not for sensor management. In this paper we try to integrate these researches to build a common architecture dealing with data fusion and sensor management providing also an application example.

2 Multi-Agent System Architecture

In this section is described an overview of the architecture and the different underlying agents involved (see Figure 1) to solve the problem of sensor management and data fusion. Also is proposed a third layer related with the user interface, that lets an eventual operator to monitor and control the state of the environment.

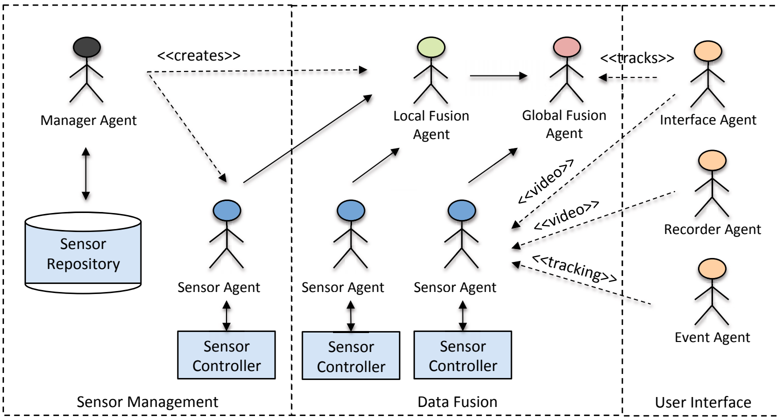

Fig. 1 Different types of agents in the architecture for solve problems of Sensor Management and Sensor Data fusion

In the sensor management side we have the Manager Agent and the Sensor Agent, that in general terms provides the management and control for the sensors presented in the system. There is only one Manager Agent in the environment, and one Sensor Agent for each camera. In the data fusion side, we have the Local fusion Agent and the Global Fusion Agent that will allow the system obtain non-redundant tracks of the different targets in the environment. All the agents are explained in more detail in the following subsections.

2.1 Sensor Agent

This agent is related with the sensor itself, and it basically provides an interface to other agents and systems to perform actions or get data from the sensor. This agent mainly depends on a software component called Sensor Controller, which is a middleware between the agent and the physical sensor. This middleware is actually developed and described in [2] and it currently provides three different interfaces: video, PTZ control, and a tracking interface which provides real-time tracking information. Thereby any Sensor Agent regarding to control a video sensor will only have to handle a common interface provided by the Sensor Controller and do not take care about the physical interfaces or protocols involved in the sensor. Depending on the underlying sensor, this agent will have different capabilities, like provide color video sequences, thermal information, PTZ control, tracks information, etc.

In the Sensor Controller middleware can be several tracking modes working at the same time depending on the features that other agents want to extract, like blob, color, and thermal tracking or face recognition. But the basic tracking information provided by the middleware is at least the presented in Table 1:

Table 1 Basic track information provided by the middleware

Value	Description
Sensor Id	The unique Sensor Id in the network
Sensor Type	The Sensor type used to get the information of the shape, like color video camera, thermal camera, infrared camera, etc.
Track Id	The Track Id provided by the tracker.
Track Type	The Track Type. It will depend on the tracking mode used and the sensor used, so may be something like color blob, face, thermal blob, etc.
Track Size	The shape size of the track itself, like width, height, etc.
Tack Shape	The shape detected like rectangle, ellipse, etc.
Track Location	The x, y coordinates of the track in the image provided by the sensor. May be referred to the center or bottom of the shape in the image.
Track Location (2)	Real world coordinates (the camera needs to be calibrated). Also is useful some kind of measurement-associated error.

2.2 *Manager Agent*

Manager Agent is the responsible of the MAS initialization in an easy way, so start the MAS architecture do not become a tedious task every time you want to achieve video surveillance. In this way, this agent relies on a Sensor Repository that contains information about all the sensors deployed. The information for each sensor is something like sensor id, sensor type, real-world sensor placement, calibration data, and the information about the Sensor Controller to employ. But not only contains information about single sensors, it also keeps information about the relationship between sensors, that is, when there exist sensors with overlapped fields of view. This information should be introduced manually as the initial setup of the architecture, thereby it can be reused in further launches of the system.

So the initial functioning of the Manager Agent can be divided in two steps. The first one is related with the dynamically creation/initialization of Sensor Agents. For each sensor present in the environment, the Manager Agent creates one Sensor Agent with the specified sensor information, so the Sensor Agent initializes the sensor and become available for other agents in the architecture. The second one relies on create static coalitions for data fusion between sensors. In this way, for each set of sensors with overlapped fields of view, it is created a Local Fusion Agent regarding tracks of the involved sensors.

2.3 *Local Fusion Agent*

This agent is dinamically created by the Manager Agent at the system startup. Its concern is to communicate with the assigned Sensor Agents to request them the tracking feature in order to achieve a local data fusion, which is a process where all the possible redundant tracks generated by overlapped fields of views can be fused to provide an unique representation of each track. This will provide to the system using this architecture a set of non-redundant tracks with all the information provided by different sensors. In addition it is useful, as is shown in the following sections, for the coordination task itself.

The proposed Local Fusion Agent will be able to perform the data fusion using the tracking information provided by each Sensor Agent, like the shape size, shape location, etc., presented in Table 1. Depending of the tracking capabilities of the Sensor Agents present in the coalition, this agent may try to use different fusion approaches. As a first step we will suppose that all the Sensor Agents are able to provide tracks with their corresponding locations in the real world, in this way, we can achieve a location-based fusion. In this way, the information that may provide the Agent as a result of the coalition is the one presented in Table 2.

Table 2 Track information as a result of a fusion

Value	Description
Track Id	An unique Track Id for the local coalition
Sensor-Track Mapping	This field will contain the relation of the underlying sensors and tracks that are contributing to generate this track, like: S1 Track 1 S2 Track 3 S3 Track 2 Also a track in the coalition may be obtained only by one single track, like: S1 Track 2
Track Location	Real world coordinates of the fused track. Also is useful some kind of measurement-associated error.
Additional Track information	As the fusion may be obtained from different kind of tracks, the idea is to provide here a general description of the combined tracks. Also it may be obtained with the Sensor-Track Mapping field.

2.4 Global Fusion Agent

Global Fusion Agent is the responsible of request and receive tracks both from Local Fusion Agents, and Sensor Agents that are not in a coalition with a Local Fusion Agent. All this information is integrated in order to provide a global view of the environment being monitored by all the Sensor Agents. It also allows obtaining a continuity of targets across the whole area covered by the sensor network. The information provided is quite similar to this one presented in Table 2, changing only the Track Id, as it will be an unique Id in the whole system.

2.5 User Interface Agents

In this group we may find different interesting agents supporting the communication with the operator to present the data and allow the management of the different surveillance agents. For example there is defined an Interface Agent that will be able to present the different video feeds and also a global vision of the tracks presented in the system through the Global Fusion Agent. Other Agent may be the Recorder Agent that can be used for retrieving video sequences of the different cameras for its recording. And finally the Event Agent that can be used along with Sensor Agents for receiving tracks in order to provide events to the operator, for example if a track enters in a field of view, or in a specific zone of the environment.

3 Application Example

After the general architecture functioning has been described, this section wants to provide an example of how this architecture can be used to control active or pan-tilt-zoom cameras. In this case is defined a PTZ Agent interacting with the architecture. So, suppose that we have a set of cameras in which three of them have an overlapped field of view, meanwhile there is another one working alone. With this configuration we can start thinking about different utilities for the PTZ Agent, both in selfish or coordinated operation modes.

In this way, when a new PTZ Agent is deployed in the architecture, it should start dealing with the Sensor Agent assigned. The PTZ Agent should check that the capabilities of the Sensor Agent allow pan-tilt-zoom controls in order to satisfy its own goals. The goals of the agent may vary depending on the operator preferences, in this work we propose different examples that can be solved with the use of the architecture.

3.1 Selfish Operation Mode

At first we should consider different kinds of PTZ controls, for example, some goals may depend on using some tracking features of the Sensor Agent, and some others may be achieved only with the PTZ feature.

- Automatic movements: the agent will turn the camera orientation automatically to monitor different areas. Can be movements like continuous or aleatory scanning, swapping between predefined areas, or follow simple predefined paths in the environment. This is the most basic and simple feature that this agent will provide using only the PTZ control.
- Target tracking: In this case this agent may request some of the tracking features available in the Sensor Agent, like simple blob tracking, face detection, or color tracking. So, with a set of tracks and the PTZ control, it can start to perform different target tracking, like maximizing the number of targets in the image, or follow the most relevant track (depending on the user preferences and the track features). Also is possible to request different tracking modes at the same time to the Sensor Agent like blob tracking and face detection and follow the tracks depending on the information given. For example, if there are only simple blobs, track them, but if a face is detected in a given moment the agent can switch to start its tracking. Also this operation mode can be combined with the previous

one, i.e, if there is no tracks in the scene, it can start looking different areas of the environment to detect them.

3.2 Coordinated Operation Mode

The previous defined examples may be useful for cameras that do not have overlapped fields of view and do not have to achieve a priori any kind of coordination. However, when we have some cameras covering the same area, we can obtain a better surveillance performance by letting the cameras get coordinated to satisfy some goal. For example, ensure that every target in the shared environment is being monitored (when possible), or each camera is tracking different targets, or many cameras collaborates to acquire as much information as possible about one single target selected by the operator, or also, not in the tracking field, all the cameras get synchronized to perform scanning of the environment.

In this way is necessary to define how to start the collaboration mechanisms and how to achieve them using the proposed architecture. In the following is described how the system may work:

- Local Fusion Agent Events: Each new PTZ Agent included in the system to control a Sensor Agent, can be joined to the Local Fusion Agent coalition in order to retrieve information about the presence of new tracks and fusions on the shared area. These events will let the agents listening to determine if there is necessary create new dynamic coalitions with other PTZ Agents to achieve collaboration. The information we propose to send in general terms is the described in table 3.
- PTZ Agents reactions: Each PTZ Agent can react in different ways to the events received, and it may depend on its current state or the personal goals to satisfy. But it is supposed that if a PTZ Agent has been registered for receive events, is

Table 3 Basic track information provided by the middleware

Event Name	Description
Track Creation	When a new track is detected in the Local Fusion (can be from any camera in the coalition), a message is sent to all the PTZ Agents listening with information presented in Table 1 to identify the track.
Track Deletion	This message is sent when some track no longer exists in the Local Fusion coalition. This message contains the information presented in Table 1.
Fusion Creation	This message is sent when a new fusion has been created in the Local Fusion coalition, that is, at least two tracks of different source sensors has been combined into one single track. Also may be sent when a new track joins to a previous existing fusion. The information sent is the information provided in Table 2.
Fusion Deletion	AThis message is sent when a fusion has been destroyed, that is, a previous existing combination of tracks no longer exists in the Local Fusion coaliton. Also may be sent when a track no longer belongs to an existing fusion. The information sent is the Fusion information presented in Table 2.

for collaborate with other agents present in the architecture. Thus we can suppose that in general the PTZ Agents can achieve the following reactions:

– Track Creation: This event may be omitted by many PTZ Agents but may be useful for alert it about the presence of new targets, and if necessary, switch the camera orientation to start monitoring them.
– Track Deletion: Similar to the track creation, this event may be used for alert that an agent has ceased to see some target, thereby any other agent in the environment can switch to the latest known target location to see if the target is available from its field of view.
– Fusion Creation: This event will notify that at least two cameras have in their fields of view at least one shared target. It may be used for different purposes, but we think that the interesting one is the creation of dynamic coalitions between the involved PTZ Agents in order to achieve coordination about the shared information. The purposes of dynamic coalitions are discussed in the following point.
– Fusion Deletion: Similar to the fusion creation, this event may induce the elimination of a dynamic coalitions between PTZ Agents or also get out some specific PTZ Agent from one coalition.

- Dynamic Coalition of PTZ Agents: The purpose of a dynamic coalition is to let the involved PTZ Agents (at least two) to collaborate in a common pursued goal. Suppose that a PTZ Agent (P1) listening in the Fusion Agent coalition receives a notification about a new fusion done with other PTZ Agent (P2). Depending on P1 goals, it may suggest to P2 create a new coalition in order to start a collaboration to satisfy a collaborative goal (suppose for this case obtain as much information about the fused target). If P2 agent accepts the coalition creation, then it is started a new temporary agent called Coalition Agent, which will be able to coordinate P1, and P2. This Coalition Agent will start to receive the tracks data from P1 and P2 from the Local Fusion Agent in order to decide or planify the tasks that should be achieved by P1 and P2 (in this case specify P1 and P2 to track the same fused target). This interaction for the dynamic coalition creation is presented in Figure 2.

  In this way, different coalitions may response to different behaviors, depending on the coalitions goals. For example suppose that we have a Coalition Agent specialized in track different targets for each camera, or another one that priorices the targets by some rule or inference and select what target should be tracked by each PTZ agent (maybe the same for all if its is visible), or also another one that avoid overlapped fields of view depending on the fusion information. Thus, there can be many examples that can be pursued using this architecture.

  This kind of dynamic coalitions that allows coordination between cameras may be started by agents (as shown in Figure 2) depending on the fusion events, or also by an eventual operator using the system, so rich coalitions with different goals can be established to improve the survelliace system. In general terms, we consider that if a PTZ Agent is actually working in a coalition, it cannot enter to collaborate to another coalition.

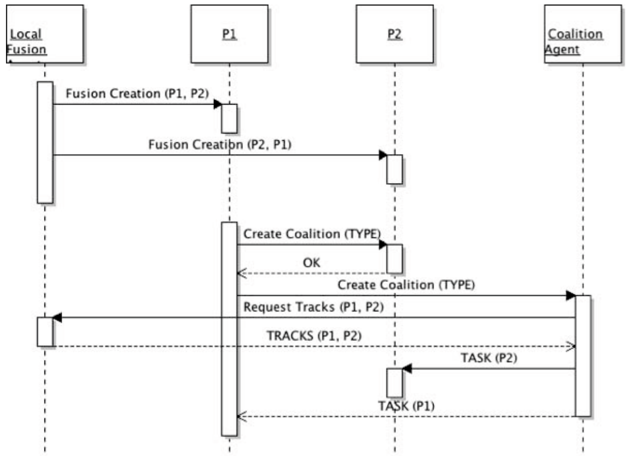

Fig. 2 Dynamic coalition formation of two PTZ agents, P1 and P2

4 Conclusions

In this work[1] we have defined a MAS architecture in order to support the distributed management of a visual sensor network, specially for PTZ devices. It supposed to deal with the common problems of data fusion and sensor management. In this way several agents where defined to accomplish different tasks. Also we define the data fusion as an integral part of the architecture that can provide mechanisms for coordination and competition between different agents. We have also presented an application example based on the architecture to understand how it can be used for different purposes. In this way dynamic coalition between PTZ Agents has emerged as a natural way of coordination.

References

1. Akyildiz, I., Kasimoglu, I.: Wireless sensor and actor networks: research challenges. Ad Hoc Networks 22(4), 351–367 (2004)
2. Bustamante, A.L., Molina, J.M., Patricio, M.A.: Multi-camera Control and Video Transmission Architecture for Distributed Systems. In: Molina, J.M., Corredera, J.R.C., Pérez, M.F.C., Ortega-García, J., Barbolla, A.M.B. (eds.) User-Centric Technologies and Applications. AISC, vol. 94, pp. 37–45. Springer, Heidelberg (2011)
3. Castanedo, F., García, J., Patricio, M.A., Molina, J.M.: Data fusion to improve trajectory tracking in a Cooperative Surveillance Multi-Agent Architecture. Information Fusion 11(3), 243–255 (2010)

[1] This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/ TIC-1485) and DPS2008-07029-C02-02.

4. Castanedo, F., Patricio, M.A., García, J., Molina, J.M.: Extending surveillance systems capabilities using BDI cooperative sensor agents. In: Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks - VSSN 2006, p. 131. ACM Press, New York (2006)
5. Collins, R.T., Lipton, A.J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt, P., Wixson, L.: A System for Video Surveillance and Monitoring. System 69(CMU-RI-TR-00-12), 573–575 (2011)
6. Garcia, J., Carbo, J., Molina, J.: Agent-based coordination of cameras. International Journal of Computer Science and Applications 2(1), 33–37 (2005)
7. Hampapur, A.: Smart video surveillance for proactive security. IEEE Signal Processing Magazine 25(4), 136 (2008)
8. Krahnstoever, N., Yu, T., Lim, S., Patwardhan, K., Tu, P.: Collaborative real-time control of active cameras in large scale surveillance systems (2008)
9. Liggins, M., Hall, D., Llinas, J.: Handbook of multisensor data fusion: theory and practice, vol. 22. CRC (2008)
10. Manyika, J., Durrant-Whyte, H.: Data fusion and sensor management: A decentralized information-theoretic approach (1994)
11. Marchesotti, L., Messina, A., Marcenaro, L., Regazzoni, C.: A cooperative multisensor system for face detection in video surveillance applications. Acta Automatica Sinica 29(3), 423–433 (2003)
12. Molina López, J.M., García Herrero, J., Jiménez Rodríguez, F.J., Casar Corredera, J.R.: Cooperative management of a net of intelligent surveillance agent sensors. International Journal of Intelligent Systems 18(3), 279–307 (2003)
13. Patricio, M.A., Carbó, J., Pérez, O., García, J., Molina, J.M.: Multi-Agent Framework in Visual Sensor Networks. EURASIP Journal on Advances in Signal Processing 2007, 1–22 (2007)
14. Singh, V.K., Atrey, P.K., Kankanhalli, M.S.: Coopetitive multi-camera surveillance using model predictive control. Machine Vision and Applications 19(5-6), 375–393 (2007)
15. Wesson, R., Hayes-Roth, F., Burge, J.W., Stasz, C., Sunshine, C.A.: Network structures for distributed situation assessment. IEEE Transactions on Systems, Man and Cybernetics 11(1), 5–23 (1981)