Analysis of Distributed Fusion Alternatives in Coordinated Vision Agents

Federico Castanedo, Jesús García, Miguel A. Patricio and José M. Molina
Applied Artificial Intelligence Group
Computer Science Department
University Carlos III of Madrid, Spain.
{fcastane, jgherrer, mpatrici}@inf.uc3m.es, molina@ia.uc3m.es

***Abstract*—In this paper, we detail some technical alternatives when building a coherent distributed visual sensor network by using the Multi-Agent paradigm. We argue that the Multi-Agent paradigm fits well within the visual sensor network architecture and in this paper we specially focus on the problem of distributed data fusion. Three different data fusion coordination schemes are proposed and experimental results of Passive Fusion are presented and discussed. The main contributions of this paper are twofold, first we propose the use of Multi-Agent paradigm as the visual sensor architecture and present a real system results. Secondly, the use of feedback information in the visual sensors, called Active Fusion, is proposed. The experimental results prove that the Multi-Agent paradigm fits well within the visual sensor network and provide a novel mechanism to develop a real visual sensor network system.**

Keywords: Multi-Agent Systems, Distributed Data Fusion, Visual Sensor Networks

I. Introduction

In this paper we report our results when building a Multi-Agent visual sensor network. Sensor networks are related to spatially distributed multi-sensor environments. In the case of camera sensors, they cope with computer vision techniques. These are known as visual sensor networks (VSNs). A distinguished feature of visual sensors, compared with other types (pressure sensors, microphones or thermometers) is the great amount of data generated, what makes necessary more local processing to deliver only the useful information represented in a conceptualized level.

One of the main advantages of using a visual sensor networks is the increase of spatial coverage. In order to have a global view of the environment under surveillance the visual sensors must be correctly deployed. The visual configuration of the sensors involves two different categories: (1) non-overlapping field of view and (2) overlapping field of view. In the case of visual sensors with overlapped field of view, redundant information could be exploited with data fusion techniques. Data fusion is related to data combination from different sources in an optimal way [1]. The aims of using data fusion techniques are to increase precision, reliability and performance of the information obtained in the visual sensor network.

Multi-Agent systems are defined by the artificial intelligence community as a cooperative network of several intelligent software agents. An intelligent software agent [2] is a computational process which has several characteristics: (1) *"reactivity"* (allowing agents to perceive and respond to a changing environment), (2) *"social ability"* (by which agents interact with other agents) and (3) *"proactiveness"* (through which agents behave in a goal-directed way).

We argue that visual sensor networks could be designed and implemented using the Multi-Agent paradigm. Therefore each node in a visual sensor network could be modeled as an agent of the Multi-Agent system and address the data fusion as a coordinated inference process. Moreover, visual sensor networks can take advantage of the Multi-Agent community research efforts.

In this paper we provide an extension of previous works related to how Multi-Agent systems could be efficiently applied in visual sensor networks [3], [4]. The main contributions of this paper compared to previous works are the evaluation of results in a different environment and the proposal of two different data fusion schemes (Passive Fusion and Active Fusion) using the same architecture. On the one hand, the Passive Fusion scheme consists of a distributed data fusion architecture where the information is locally processed in each node and then fused into another node. On the other hand, the Active Fusion provides feedback information based on the information taken from other overlapping sensors. This feedback information allows a reasoning process in the local node in order to correct possible deviations.

Therefore, in this work, we assume an overlapping visual sensor configuration and focus on several ways of fusing the information generated from multiple visual sensors into a global one using a Multi-Agent system.

The outline of the paper is organized as follows: The next section describes related works in the area of visual sensor networks. Then, the visual sensor Multi-Agent architecture is presented in section III. Section IV discusses different approaches of the distributed data fusion. Data fusion experimental results are presented in section V. Finally, the conclusions are given in section VI.

II. Related Works

One of the first attempts in the literature to build a Multi-Agent approach for data fusion, was published in [5]. The authors proposed a Multi-Agent approach based on the blackboard model. The blackboard model is one of the first Multi-Agent communication models. An approach based on information-subscription coordination model is used in the European project MODEST [6]. The traffic surveillance system of MODEST deployed four cameras along a bridge in Brussels and it is FIPA [7] compliant. The coordination mechanism uses the directory facilitator (DF) of the FIPA platform [8]. In this project they proposed an extension of the Semantic Language (SL) so that they could take into account uncertainty and MPEG-7 descriptors. In Graf and Knoll work [9] they distinguish between two types of agents: masters and slaves which are connected using a contract net. The contract net protocol [10] is one of the most extended coordination paradigms developed in distributed artificial intelligence. However the drawback of their approach is that they use a specific communication language (not a standard one) which makes difficult interoperability with other systems. The work from P.K.Biswas et.al [11] is also based on the multi-agent paradigm. However they proposed a mobile-agent-based approach where the base station deploys mobile agents that migrate from node to node and fuse the information locally in each node. Orwell et. al. [12] describes an architecture to implement scene understanding algorithms in the visual surveillance domain. The main objective is to obtain a high level description of the events observed by multiple cameras. In their architecture each camera is associated to an agent. They also create one agent per each object detected in the scene. The tracking is performed on the ground plane. Although, they describe a data fusion trajectory they do not use a standard multi-agent architecture paradigm. Agent-based software systems can be also applied to the management of a set of networked distributive sensors as in [13], where a hierarchical agent-based network is proposed.

III. Multi-Agent Visual Sensor Network Architecture

In the last years, many Multi-Agent languages and frameworks have been developed [14]. Our proposed architecture is based on the open source framework Jadex [15]. Jadex [15] is a Belief-Desire-Intention (BDI) Multi-Agent model. The BDI model provides a way to conceptualize the system and structure its design [3].

Our visual sensor network architecture is built by using three different types of agents and it is described in [3], [4]. Each agent has its own responsibilities and cooperate each other in order to make a coherent distributed data fusion.

A. Architecture

We could briefly review, the three different types of agents as follows:

1) *Surveillance-sensor agent:* This type of agent tracks all the objects and sends the data to a fusion agent. It acquires the environment information through a camera and performs the local signal processing side. The tasks carried out are: detection of objects, data association, state estimation, projection on fusion coordinates and the communication with the fusion agent.
2) *Fusion agent:* It performs the fusion of the data received from each surveillance agent. Then they fuse the information received from the surveillance-sensor agents, which is received in FIPA ACL messages and it is time stamped.
3) *Interface agent:* This agent receives the fused data and shows it to the final user. It is also the user interface of the surveillance application.

B. Fusion Process

For each detected target, the surveillance-sensor agents internal tracking provides an associated track vector of features $\hat{X}^{S_i}_{T_j}[n]$, containing the numeric description of their features and their state: location, velocity, dimensions, etc. and associated error covariance matrix, $\hat{P}^{S_i}_{T_j}[n]$. A specific parameter (*Looking Interval* in milliseconds) sets the frequency when sending the objects detected under the field of view. The fusion agent receives the track information from the surveillance-sensor agents through a TCP/IP network using FIPA ACL messages. The most important fusion agent parameters involved in the fusion process are:

1) *Temporal Difference:* It is a value in milliseconds which is used to discriminate when the measurements are from different tracks.
2) *Spatial Difference:* It is a parameter in centimeters which specify a threshold used to discard a track due to a spatial inconsistency regarding the others tracks.
3) *Feedback Frequency:* It is a value in milliseconds, used in the active fusion which indicates the frequency of feedback messages.
4) *Fusion Frequency:* This parameter sets the frequency in the fusion process. Every *fusion frequency* milliseconds the fusion process is performed by the fusion agent.
5) *Fusion Type:* It indicates the fusion type: active fusion (with feedback) or passive fusion (without feedback).
6) *Fusion Algorithm:* It establishes the fusion algorithm used.

In the case of Active Fusion, when surveillance sensor agents receive the feedback messages, a decision process is carried out in order to correct the possible deviations. The main surveillance-agent parameters involved in the Active Fusion process are:

1) *Feedback Threshold:* It establishes the number of feedback messages an agent takes into account in order to reason about the quality of the numeric descriptions of the features.
2) *Spatial Difference:* It is a value in centimeters which specifies a threshold to detect inconsistent measurements with respect to the fused values.

IV. Distributed Data Fusion Alternatives

The presence of multiple data sources and fusion nodes provides many possibilities in the architecture [16]. In the data fusion literature three different types of distributed architectures are widely adopted. In the next sections we present how the proposed Multi-Agent architecture fits on these types of architectures.

A. Passive Fusion

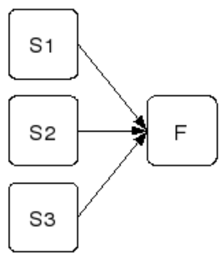

Figure 1. Passive Fusion scheme

In this type of fusion scheme (see Figure 1) each surveillance-sensor agent processes the information of the environment in their field of view and sends the tracks to the fusion agent; that is what we call Passive Fusion. The fusion agent is in charge of performing the data fusion by using the received tracks. In that way, a specific value in milliseconds (*Fusion_Frequency*) to cyclically activate the data fusion process is used. As the tracks are time-stamped we can compare the time of two different tracks, so out of sequence measurements could be treated. The internal clock of the machines where each surveillance-sensor agent runs, must be also synchronized with the others clock machines. Therefore every *Fusion_Frequency* milliseconds the fusion agent builds a fused track with the received information.

B. Active Fusion

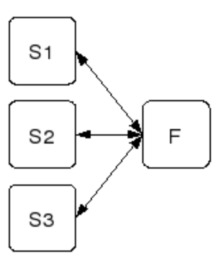

Figure 2. Active Fusion scheme

The idea behind this data fusion scheme is to provide feedback information to each surveillance-sensor agent involved in the fusion process. This feedback information allows each surveillance-sensor agent to reason about the quality of the information which is being sent to the fusion agent in relation to the other overlapped sensors. As each surveillance-sensor agent is autonomous, it can decide about the inconsistencies in the information and correct them before they are sent to the fusion agent. This process involves an alignment of the information in order to obtain a coherence of the reason process by means of a cooperative mechanism. Therefore, Active Fusion implies that each surveillance-sensor agent is able to correct values, delete objects and change projections.

In this type of architecture the fusion process should deal with data incest. Data incest refers to the inadvertent multiple use of raw measurements several times as though they were independent which can lead to biases in estimates and over confidence in their accuracy. Data incest risk with this scheme of active fusion is moderated, as feedback information can be used first to correct local tracks, which are used later to obtain the fused result in next fusion iteration.

C. Peer to Peer Fusion

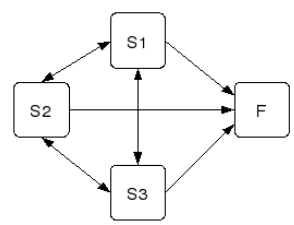

Figure 3. Peer to Peer Fusion scheme

In this type of data fusion scheme each surveillance-sensor agent can communicate with any other agent and the communication is asynchronous. The fusion is performed inside each surveillance-sensor agent and then it is sent to the fusion agent. An important drawback of this coordination scheme is the communication overload. The usual solution is to create subgroups of surveillance-agents which are managed by a fusion agent.

In this data fusion scheme, data incest must be considered further since it can appear more frequently than in previous scheme. The systematic presence of data incest is caused by cyclic paths in which the information recirculates from output of a fusion node back to the input of others [17]. One of the main effects is an optimistic estimation of the error made in the fused result. McLaughlin et. al [18], [19] developed a data incest removal strategy for distributed architectures. They also proposed a Bayesian network model to deal with data incest [20]. Covariance Intersection (CI) [21] is a data processing solution to the problem of data incest in multi-cyclic communication networks. CI solves the problem of correlated inputs, but it is undefined for inconsistent inputs. To solve this, Uhlmann developed Covariance Union (CU) [22].

V. Experimental Results

Distributed data fusion usually involves two problems: (1) associating the tracks generated from the different nodes to determine if they correspond to the same targets and (2) combining the information for those tracks that are associated

[16]. In order to solve the first problem, in these experiments we assume that there is only one track. However a system must be able to track more than one target, but this is not the scope of the paper. For the second problem we carry out experiments using the Passive Fusion scheme, where each surveillance-sensor agent performs the video analysis algorithms and sends the generated tracks to the fusion agent. The fusion agent performs the data fusion process with the tracks sent by the surveillance-sensor agents. The fusion algorithm used in the Passive Fusion coordination scheme was already published in [4] (it is based on a selection/combination scheme).

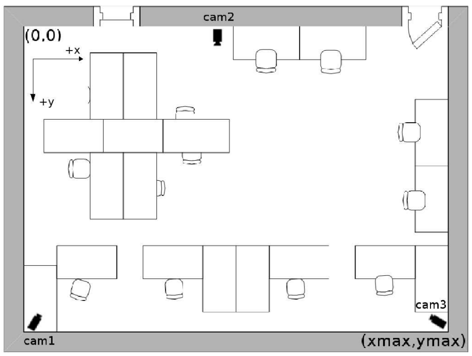

Figure 4. Experimental environment

An indoor evaluation of the proposed architecture is carried out in our laboratory. In this laboratory, three Sony EVI-100 Pan-Tilt-Zoom (PTZ) cameras are each one connected to a Matrox Morphis frame grabber. Figure (Fig. 4) illustrates the camera layout in this experimental environment. The positions of the three cameras and the calibration can be observed in the axis reference. Each PTZ camera is controlled by a surveillance-sensor agent which runs in different dedicated computers. There is also a fusion agent (running in pc under linux) which receives surveillance-sensor agent's tracking information. The computers are connected by a 100 Mb ethernet connection.

A record of 300 frames (768x576 pixels) from one person randomly moving (in a zone of 660cm x 900cm) using three different cameras is performed. Each of the frames was handmade annotated in order to obtain the ground-truth, which is the real tracking value to be obtained. The ground-truth tracking values was obtained as follows: we annotated the local coordinate of all the frames (300 frames x 3 cameras) where the person is viewed (e.g. see Figure 5), then the calibration function is applied and finally we compute the average between each frame value of the different cameras. Obtaining the ground-truth tracking is a tedious task, but it allows us to obtain a quality reference.

In figure (Fig. 10) the input images from frame 75 are shown. In this experimental indoor scenario a lot of occlusions occurs which affects each local tracking.

Figure 5. Ground-truth value in frame 60 from camera 1 (464, 314) in local coordinates (white point)

A. Passive Fusion Experiments

1) Normal Conditions: In order to evaluate the Passive Fusion coordination scheme, we deploy three surveillance-sensor agents and one fusion agent. Each surveillance-sensor agent process one video record and the *Looking Interval* parameter was a random delay, uniformly distributed between 0 and 100 milliseconds ($\approx$ 10 frames per seconds). The fusion agent runs with these parameters: 300 milliseconds for *Fusion Frequency*, 300 milliseconds for *Temporal Difference* and 20 centimeters for *Spatial Difference*. Since the *Looking Interval* introduces randomness in the system, ten experiments were conducted and the average of the results is plotted. In figure 6 we show the fused tracking positions of the horizontal values over the time and then we compare them to the ground-truth previously obtained (in that frames where we have ground-truth values). The same analysis is shown in figure 7 but for the vertical values.

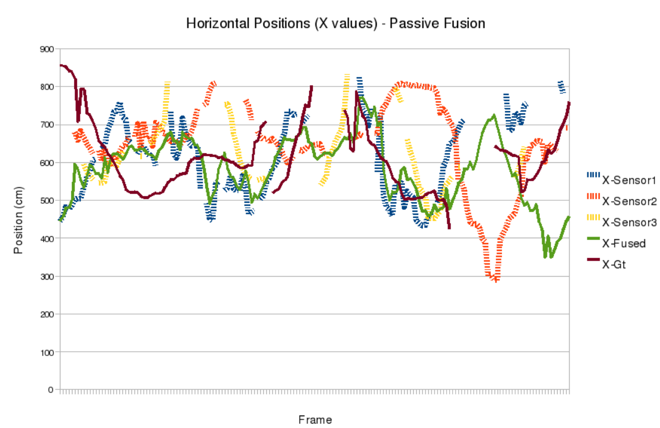

Figure 6. Horizontal tracking positions (x values)

2) Abnormal Conditions: In this experiment we introduce a deviation in one of the sensors (sensor 2) which consist of a systematic error of 30 centimeters in the common values of the tracking positions. As we expected, the fusion process discards some sensor 2 tracks due the *Spatial Difference* constraint (since it was set to 20 cm). The parameters of the experiments

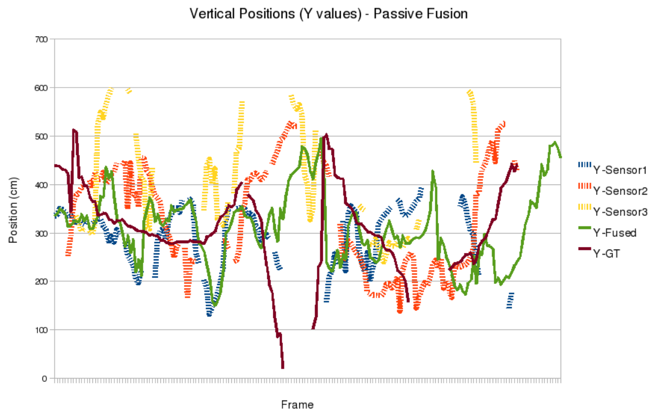

Figure 7. Vertical tracking positions (y values)

were the same as the previous one. The fused tracking results are shown in figure 8 and 9. The fused values obtained have more jitter than the previous results, mainly in the vertical values.

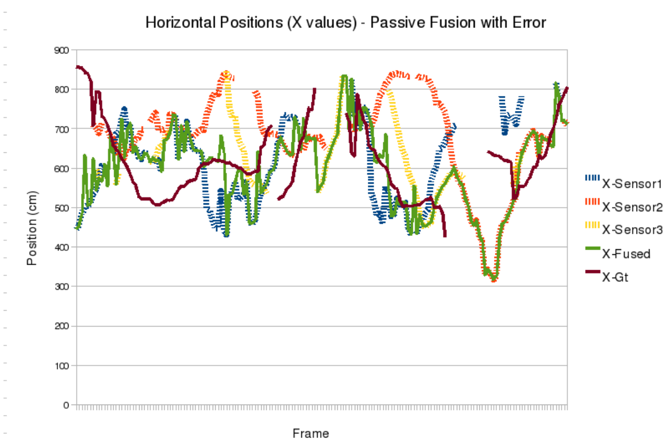

Figure 8. Horizontal tracking positions (x values) with a systematic error of 30 centimeters in sensor 2

VI. Conclusions and Future work

In this paper we present an analysis of an implemented Multi-Agent system operating in real situations. We specially focus on the distributed data fusion process between the agents involved. Three different types of distributed data fusion architectures have been proposed and implemented through coordinated behaviors. Experimental results of Passive Fusion are presented. These results show that the Multi-Agent paradigm fits well in a visual sensor network architecture. Moreover existing data fusion techniques could be implemented by using Multi-Agent systems and therefore it would provide more flexibility than classical systems. When the fused information is affected by systematic errors, these errors could be corrected by using the proposed Active Fusion scheme, which is ongoing work.

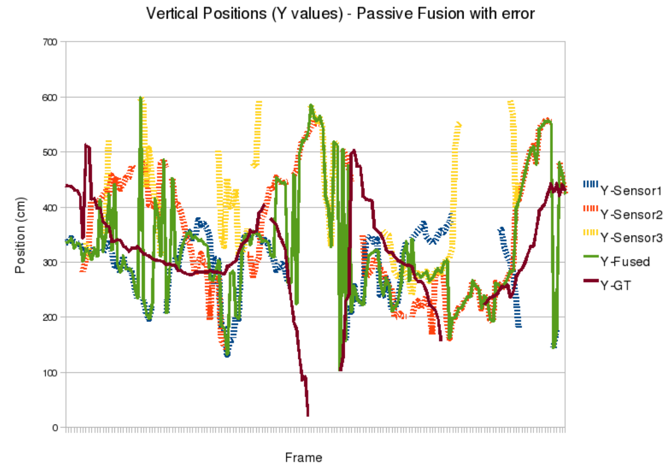

Figure 9. Vertical tracking positions (y values) with a systematic error of 30 centimeters in sensor 2

Acknowledgment

This work was partially supported by projects MADRINET, TEC2005-07186-C03-02, SINPROB, TSI2005-07344-C02-02.

References

[1] E. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech House Inc, Norwood, Massachussets, U.S, 1990.

[2] M. Wooldridge and N. Jennings, "Intelligent agents: Theory and practice," *The knowledge Engineering Review*, 1995.

[3] F. Castanedo, M.A. Patricio, J. Garcia and J.M. Molina. "Extending surveillance systems capabilities using BDI cooperative sensor agents," *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pp. 131–138, 2006.

[4] F. Castanedo, M.A. Patricio, J. Garcia and J.M. Molina. "Robust data fusion in a visual sensor multi-agent architecture," *10th International Conference on Information Fusion*, pp. 1–7, 2007.

[5] V. Berge-Cherfaoui and B. Vachon, "A multi-agent approach of the multi-sensor fusion," *Fifth International Conference on Robots in Unstructured Environments*, pp. 1264-1259, 1991.

[6] "Multimedia Object Descriptions Extaction from Surveillance Types. http://www.tele.ucl.ac.be/PROJECTS/MODEST/index.html."

[7] FIPA, "Fipa communicative act library specification," 2001.

[8] L. Botelho, R. Lopes, M. Sequeira, P. Almeida, and S. Martins, "Inter-agent communication in a FIPA compliant intelligent distributed dynamic-information system," *Proc. 5 thInternational Conference on Information Systems Analysis and Synthesis (ISAS99)*, 1999.

[9] T. Graf and A. Knoll, "A Multi-Agent System Architecture for Distributed Computer Vision," *International Journal on Artificial Intelligence Tools*, vol. 9, no. 2, pp. 305–319, 2000.

[10] R. Davis, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 29, pp. 1104–1113, 1980.

[11] P.K. Biswas, H. Qi and Y. Xu. "Mobile-agent-based collaborative sensor fusion," *Information Fusion*, 2007.

[12] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G. A. Jones, "A multi-agent framework for visual surveillance," in *ICIAP '99: Proceedings of the 10th International Conference on Image Analysis and Processing*, (Washington, DC, USA), p. 1104, IEEE Computer Society, 1999.

[13] Q. Zhu, S.L. Aldridge and T.N. Resha, "Hierarchical Collective Agent Network (HCAN) for efficient fusion and management of multiple networked sensors." *Information Fusion* , pp. 266-280, vol. 8, no. 3, 2007.

[14] V. Mascardi, D. Demergasso and D. Ancona. "Languages for Programming BDI-style Agents: an Overview," *WOA*, pp. 9–15, 2005.

Figure 10. Input Images (frame 75) of camera 1, 2 and 3. From top to bottom, camera 1, camera 2 and camera 3.

[15] A. Pokahr, L. Braubach and W. Lamersdorf, "Jadex: Implementing a BDI Infraestructure for JADE Agents," *Search of Innovation (Special Issue on JADE)*, vol. 3 pp. 76–85, September, 2003.

[16] M.E. Liggens, C.Y. Chong, I. Kadar, M.G. Alford, V. Vannicola and S. Thomopoulos. "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol. 85m pp. 95–107, 1997.

[17] A.D. Marrs, C.M. Reed, A.R. Webb and H.C. Webber, "Data Incest and Symbolic Information Processing." *technical report, United Kingdom Defence Evaluation and Research Agency*, Mar. 1999.

[18] S. McLaughlin, R.J. Evans,and V. Krishnamurthy, "Data incest removal in a survivable estimation fusion architecture" *Proceedings of the Sixth International Conference of Information Fusion*, 2003.

[19] S. McLaughlin, V. Krishnamurthy and S. Challa, "Managing data incest in a distributed sensor network." *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'03)*, 2003

[20] S. McLaughlin, V. Krishnamurthy and R.J. Evans, "Bayesian network model for data incest in a distributed sensor network." *Proceedings of the 7th International Conference on Information Fusion*, 2004, pp. 606–613.

[21] S.J. Julier and J.K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations ." *Proceedings of the American Control Conference*, 1997.

[22] J.K. Uhlmann, "Covariance consistency methods for fault-tolerant distributed data fusion." *Information Fusion*, pp. 201–215, vol. 4, no. 3, 2003.