Bustamante, Álvaro Luis, Molina, José M., Patricio, Miguel A. (2014). Information fusion as input source for improving multi-agent system autonomous decision-making in maritime surveillance scenarios. *Proceedings. FUSION 2014: 17th International Conference on Information Fusion, Salamanca, 7-10th July 2014,* [8 p.]

URL: https://ieeexplore.ieee.org/document/6916018

Information fusion as input source for improving multi-agent system autonomous decision-making in maritime surveillance scenarios

Alvaro Luis Bustamante   Jose M. Molina   Miguel A. Patricio

Decision making problems are usually referred as a cognitive process resulting in the selection of an action among several alternatives. Autonomous entities taking these actions like multi-agent systems, needs to process and understand its environment state to frequently update its beliefs, and then, select an optimal action. As an environment can be composed by several sources of information, it is useful for a multi-agent system, a way to process integrated information of multiple data which represents the same real-world object. This information can improve the agents knowledge and let select better actions than processing simple raw data. Most information fusion research has had a technical and algorithmic focus, and takes little attention to high level decision making, although some studies relate fusion to human decision making. However, in this paper is proposed the use of fused information as an input source for supporting and improving the decision making capabilities of autonomous agents in maritime surveillance scenarios.

I. INTRODUCTION

Maritime Surveillance systems, generally conducted by military and law enforcement agencies, are commonly used for identifying and intercepting threats in seaports, coastal areas, maritime boundaries, maritime platforms, or important installations. Monitoring extensive environments, like the maritime, requires a wide deploy of monitoring sensors detecting targets of interest. In such deploy uses to be both collaborative and non-collaborative sensors like Automatic Identification System (AIS), radar, and Closed-circuit Television (CCTV) [1]. It is common to obtain all this distributed information in a common base station and use a Vessel Traffic Service (VTS) for its monitoring in real-time and evaluate possible threats.

AIS is a collaborative sensor integrated in each vessel that periodically reports the vessel's identity, current position, speed, bearing, and other useful information for its monitoring. Radar in contrary is a non-collaborative sensor that periodically checks the environment using radio waves for detecting targets. Those sensors are disposed depending on the specific environment conditions, and it is common to use several sensors covering different areas. Using several or complementary sensors, like radar and AIS, may dispose overlapped monitoring areas, causing the same target being observed and reported by more than one sensor, obtaining redundant and not integrated information. This is a problem widely explored in the multi-sensor fusion community [2], and integrated in real surveillance systems [3]. The benefits of using data fusion are clearly demonstrated and any state-of-the-art VTS should include a data fusion system integrating sensors information.

Some systems also includes ultra long range surveillance cameras, usually placed in scenarios where security is a great concern, which is growing in the last years as maritime piracy continues [4], threatening vessels, goods, and crew. These pan-til-zoom (PTZ) cameras are usually controlled manually by human operators for monitoring important events and vessels. Some of them can be integrated with external sensors like radars for automatically start monitoring new vessels that enters in a given zone [5]. Those approaches uses to fail in large deployments where several sensors and cameras are available and its control can be a challenging task.

In this paper it is explored the possibility of automatically controlling PTZ cameras according to VTS operator preferences. There are many research focused controlling PTZ cameras depending on environment perceptions, like the works described in [6], [7], [8], and [9]. None of these approaches contemplates the use of external sensors for improving PTZ control. As there are many sources of information on maritime surveillance scenarios, it can be useful to use all available resources. Moreover, it is discussed how by using available data fusion systems, it is also possible to improve the overall automatic control performance.

For automatically controlling PTZ cameras we are extending the work proposed in [6], [7], that describes a Multi-Agent System (MAS) architecture for distributed PTZ control. The MAS architecture is mainly composed by Camera Agents that are monitoring the environment through the cameras. There are also Fusion Agents that integrates cameras information allowing different Camera Agents to collaborate. In this previous works it is not mentioned the use of external sensors as sources of information, so we extend this work in the way that there are new surveillance sensors collaborating. Not only is discussed the use of different sensors, it is also evaluated how using data fusion can improve the decision-making process achieved by the MAS.

Achieving this evaluation can be a complex task in the maritime environment, so it is proposed the use of a simulation tool that helps designing maritime scenarios by simulating radar sensors, AIS Stations, PTZ cameras, vessels with custom trajectories, sensors detections, data fusion, agents, and so on. All this elements can be easily placed over a map representation and simulate its behavior in real-time. Using this tool will allow comparing agents performance when using different sources of information and using the same decision-making process.

The rest of the paper is organized as follow: section II provides an environment description outline, identifying the different elements involved. Section III describes the decision-making process achieved by agents for selecting best monitoring target among all the alternatives. Section IV describes the simulation tool and the experiments achieved in other to prove that using data fusion improves the system performance in terms of effective monitoring, redundancy, and lost time. Finally there are provided some conclusions and future works.

II. Environment Description

As stated in the introduction, this work is focused on the maritime surveillance environment, with a MAS controlling a set PTZ Cameras. Depending on the information provided by all environment sensors, the MAS should control PTZ cameras, for automatically start monitoring the most relevant targets according to operator preferences. This reduces work to operators when controlling several PTZ cameras which requires to be coordinated.

This scenario is briefly outlined in figure 1. In such figure there are disposed different sensors like Radar, AIS, and PTZ cameras. Over all those sensors, there is a multi-agent system receiving environment perceptions and sending control signals to PTZ cameras. The data fusion system is also listening information provided by different sensors, so it can be also contemplated as a new source sensor providing integrated information to MAS. Thus, the MAS is the responsible of analyzing all incoming perceptions, reasoning about the environment state and operator preferences, and finally, taking actions over PTZ cameras.

The data fusion system design is out of the scope of this paper, but it follows a similar architecture like the work described in [3]. The MAS architecture used in this approach is almost the same as this described in [6], [7], with the difference that this work contemplates using new sensors and a data fusion system integrating all its information.

III. Muti-Agent System Decision-Making

Decision-making for a MAS can be regarded as a cognitive process that selects an action about several alternative possibilities. Each decision-making process should produce a final choice. In this scope, the alternatives are the different available tracks to be monitored, and the choice, the track selected to be monitored (Track term is used in this paper

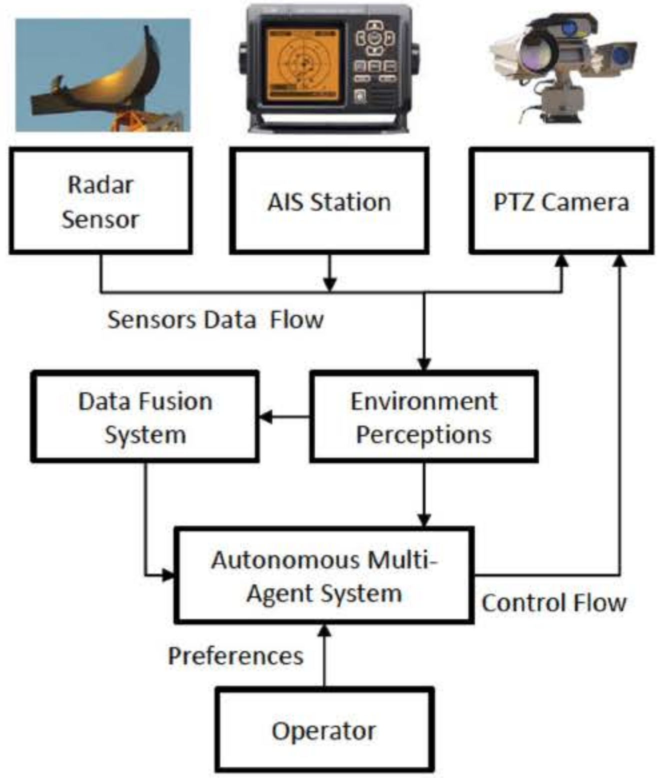


Fig. 1: Radar, AIS and PTZ cameras, and even data fusion are sources of information for multi-agent system. Those perceptions are processed to automatically control pan-tilt-zoom cameras according to operator preferences.

to refer targets at sensor tier). There is always a trade-off in this kind of problems, since monitoring a given track will prevent monitoring another interesting tracks. In this way there is important to establish priorities about what track to be monitored in each moment.

There are several alternatives for achieving Multi-Criteria Decision-Making (MCDM) in a multi-agent system like those described in [10], [11]. For testing how the data fusion can improve the decision-making process, we have selected the Analytic Hierarchy Process (AHP) [12], as it provides a comprehensive framework for structuring a decision problem, for representing and quantifying its elements, for relating those elements to overall goals, and for evaluating alternative solutions. In general terms, a domain expert like an VTS operator can easily establish its qualitative priorities.

In a simplified AHP hierarchy it is required to select a goal, define a criteria, and present alternatives for its evaluation. In this case the goal is to select a monitoring track among all the different alternatives present in the environment in terms of tracks reported by sensors. The criteria selection and their priorities should be established according to operator preferences. The example hierarchy selected for representing this problem is illustrated in figure 2 where the top goal, criteria, and alternatives are presented in a two levels AHP.

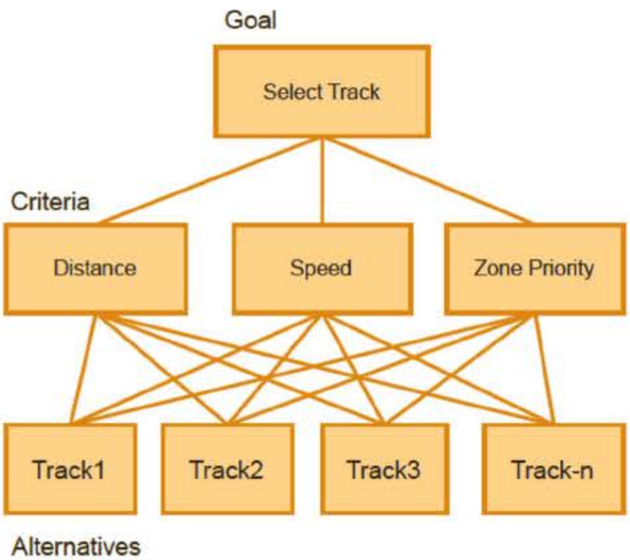


Fig. 2: Simplified Analytic Hierarchy Process selected for evaluating fusion impact on decision-making performance. The top goal is to select a track among all the alternatives based on established criteria: Track distance to controlling camera, track speed, and track zone priority

In the AHP framework, criteria priorities are established by creating positive reciprocal matrices, also known as pairwise comparison matrices [12], which establishes weights between criteria. The selected criteria resides on distance between track and camera, track speed, and zonal priority. An example of the selected weights can be summarized in matrix (1).

$$PairwiseComparisonMatrix(PCM) = \begin{array}{l} \\ DISTANCE \\ SPEED \\ ZONE \end{array} \begin{array}{c} \begin{array}{ccc} DISTANCE & SPEED & ZONE \end{array} \\ \begin{bmatrix} 1 & 2 & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{3} \\ 2 & 3 & 1 \end{bmatrix} \end{array} \tag{1}$$

Values of pairwise comparison matrix are established according to operator preferences between criteria, with values ranging from 1 to 9, which 1 means equal importance, and 9 absolutely more important [12]. AHP also allow evaluate criteria inconsistency by calculating consistency index (CI) based on the principal Eigen value for later generate a consistency ratio (CR). CR is acceptable when it is under 10%. Those indicators are specially useful when there are several criteria involved and it is hard to qualitatively adjust their weights. The CR for the proposed example is under 1%, so the inconsistency is assumable.

Based on the pairwise comparison matrix generated manually by the operator, AHP estimates a criteria priority vector. As stated by Saaty in [13] it is required to compute the principal normalized Eigen vector of the pairwise comparison matrix. The resultant normalized Eigen vector is then used as criteria priority vector. In the following it is presented the priority vector computed from *PCM* (2). It is noticeable how the zonal priority obtains a higher priority than speed or zonal criteria, as expected when looking to *PCM*.

$$CriteriaPriorityVector(CPV) = \begin{array}{l} DISTANCE \\ SPEED \\ ZONE \end{array} \begin{bmatrix} 0,297 \\ 0,163 \\ 0,540 \end{bmatrix} \tag{2}$$

In the third AHP level are present the alternatives. They also need to be compared in a pairwise comparison matrix for each criterion, and then obtain a priority vector from a normalized Eigen vector. In order to simplify the problem of assigning qualitative priorities from quantitative information like distance, speed, etc., we will compute the priority vector directly by weighted normalization instead of Eigen vector. For this case it is assumed that the distance has higher priority when it is smaller (the track is closer to the camera) and it is used (4), the speed has more priority as it is bigger (anchored or slowly tracks will be penalized) (3), and zonal priority is bigger as it increases (3).

$$w_i^{'} = \frac{w_i}{\sum_{i=1}^{n} w_i} \tag{3}$$

$$w_i^{'} = \frac{(\sum_{i=1}^{n} w_i) - w_i}{\sum_{i=1}^{n} w_i} \tag{4}$$

Those computed priority vectors for each criterion can be composed in a single priority matrix like this shown in (5). Each column represents a criterion and each row represent an alternative (tracks in this example). Each $APM_{i,j}$ specifies then a weight for the given track i for a criterion j.

$$AlternativesPriorityMatrix(APM) = \begin{array}{c} \begin{array}{ccc} DISTANCE & SPEED & ZONE \end{array} \\ \begin{bmatrix} DistanceWeight_1 & SpeedWeight_1 & ZoneWeight_1 \\ DistanceWeight_2 & SpeedWeight_2 & ZoneWeight_2 \\ \dots & \dots & \dots \\ DistanceWeight_n & SpeedWeight_n & ZoneWeight_n \end{bmatrix} \end{array} \tag{5}$$

Finally the alternatives priority matrix APM and the criteria priority vector CPV are multiplied together, obtaining the overall priority vector for all the evaluated tracks (6).

$$TracksPriority(TP) = APM * CPV \tag{6}$$

The resultant tracks priority vector can be regarded as some kind of overall ranking between all evaluated tracks. Selecting the one with more weight, will return the most priority track according to established operator preferences. Each MAS agent system should achieve this process independently each time it requires to select the best track for monitoring

A. Solving Conflicts Between Agents

As agents calculates their track priorities independently, it is possible that more than one agent become interested in monitoring the same target. This will reduce efficiency in the MAS system as there are redundant monitoring while agents could spend its time monitoring other lower priority targets or scanning different areas.

In this way it is necessary to deal with this problem by applying some collaborative algorithm between agents. There are many research done in collaborative MCDM. For keeping the MAS as distributed as possible, we have opted for using a simple bidding system between agents. So each agent interested on monitoring a new target, should notify its intentions to other agents prior to start. If there are no other agents in conflict (they are not monitoring, or trying to monitor the same target), the agent can start it task.

However, if there is one or more agent in conflict, a collaborative process is started for deciding which agent is the winner. The winner can continue monitoring the target and the loser should carry out another task (select another target or keep looking other areas). This agreement expires after some period, so agents must revalidate their bids periodically.

IV. Experimentation

In this section is evaluated the suitability of using data fusion information in a maritime surveillance scenario for improving MAS decision-making. It starts describing the simulation tools employed in those experiments, and finally there are proposed some evaluation scenarios.

A. Simulation Tools

Evaluating performance and suitability for a MAS controlling security cameras in a real maritime scenario is a challenging task task. There are several sensors to deploy which requires a great infrastructure, access to existing monitoring scenarios could require special permissions, information captured uses to be confidential, is not usual to have such deployments nearby, and so on. There is also another question that is still more important: experiments repeatability. In order to compare different problem approaches or changes in algorithms it is necessary an environment that can repeat the same conditions along time. So different algorithms can be compared under the same exact conditions.

For this purpose we have chose simulating all those elements developing a maritime surveillance simulation tool. This tool is called VTS Surveillance Simulator, and it allows designing scenarios with Radar and AIS sensors, vessels with their trajectories, PTZ cameras, and some other map utilities for representing polygons, angles, distances, and so on. Some of the main features of this tool are briefly described below:

- Vessels: Vessels are anchored at some given point, and can optionally define ideal trajectories as a set of waypoints. Each waypoint is defined as latitude and longitude coordinates, speed, and angular speed (if there is a manouver for this waypoint). It can also define if the vessel contains an AIS transponder with its MMSI and differential sensor position. Moreover, it is possible to configure starting trajectories delays in any waypoint, so it is easier to fine adjust trajectories, stops, and crossing distances between vessels.
- Sensors: There are three types of sensors available. Radar, AIS stations, and cameras. Each sensor can define its own parameters aside its position, like period, reach, azimuth precision, and distance precision for Radar. Cameras contains several parameters to simulate a pinhole camera model, like minimum and maximum horizontal and vertical field of view, minimum and maximum focal length to simulate zoom, and also other internal parameters like sensor size. AIS stations are simpler to configure as it only requires position and maximum reach.
- Distances and Angles: It is so useful monitoring distances and angles between vessel trajectories, vessels and sensors, and other custom locations. So it is possible to place rulers to measure distances and angles between elements.
- Polygons and Polylines: Such elements are useful for defining regions of interest, assigning zonal priorities, monitoring paths, etc.
- Agents: This tool also integrates mechanisms for virtually deploy different types of agents that can receive information from sensors, communicate with other agents, and act over sensors. For this custom environment, possible agents are PTZ Agents and Fusion Agents.

All the above configurable items are not passive entities that are displayed over a map. They can be simulated in real-time to evaluate the environment. So a vessel will describe its trajectory according to configured dynamics, a Radar will generate vessels plots according to its period and report their tracks, AIS stations will capture AIS messages virtually reported from vessels depending on their speed, Agents will control cameras according to their goals and environment state, Cameras will represent their current field of view, and so on.

For further information there are available some videos[1] presenting the simulation tool in real time. An example of this tool is also available in the scenario description used in the experimentation section 3.

B. Testing Scenario

The proposed scenario is located at Santander seaport from Spain. It contains two Radar Sensors and one AIS Station. There are also three cameras located at similar places than sensors. Each camera is controlled by one independent agent, and its default behaviour is to panning within a predefined monitoring path until it finds tracks to monitor. The MAS goal is controlling cameras for tracking targets in the environment with the minimum redundancy. The minimum redundancy

[1] Video1: https://youtube.com/watch?v=ytqfizjD-vU Video2: https://youtube.com/watch?v=gWcTciCGMzI

TABLE I: Sensors description used for the proposed simulated scenario.

Sensor	Type	Description
Radar1	Radar	Max Reach 3000m, 3s period. Located at $43.4272°, -3.8060°$ (WGS84)
Radar2	Radar	Max Reach 3000m, 2s period. Located at $43.4585°, -3.7756°$ (WGS84)
AIS Station	AIS	Max Reach 5000m. Located at $43.4354°, -3.7853°$ (WGS84)
Camera1	PTZ Camera	Max Reach 3000m. Located at $43.4579°, -3.7732°$ (WGS84)
Camera2	PTZ Camera	Max Reach 3000m. Located at $43.4291°, -3.8096°$ (WGS84)
Camera3	PTZ Camera	Max Reach 3000m. Located at $43.4336°, -3.7858°$ (WGS84)

TABLE II: Agents deployed in the simulated scenario.

Agent	Type	Description
Agent1	PTZ	Controlling: Camera1. Sources of Information: Depending on experiment
Agent2	PTZ	Controlling: Camera2. Sources of Information: Depending on experiment
Agent3	PTZ	Controlling: Camera3. Sources of Information: Depending on experiment
FusionAgent	Fusion	Sources of Information: Camera1, Camera2, Camera3, Radar1, Radar2, and AIS Station

Fig. 3: Sample environment for testing data fusion for helping decision-making process. There are two radar sensors (range in white), and one AIS Station (range in black). Three pan-tilt-zoom cameras are deployed in the environment (its current field of view in blue). In red are defined some monitoring paths for each agent.

means here that two or more agents should not being monitoring the same target, so it is maximized the number of different targets monitored and the covered area.

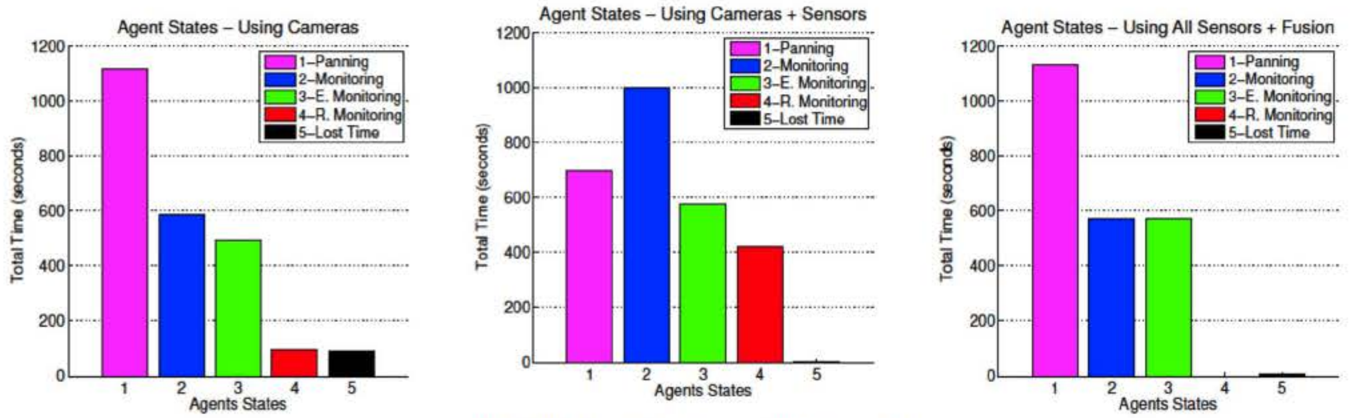
Those different sensors and agents deployed are briefly described in tables I and II, and also a map representation is available at figure 3. As can be noticed, radar sensors, with a max range contour in white, overlaps in a given region. The AIS Station, with a max range contour in black, practically covers both radar sensors range. With a maximum reach of 3000 meters for cameras, there are also possible overlapped fields of view between them. So using data fusion in this context will allow MAS to avoid tracking redundancy, and then improve the overall system performance.

The evaluation process will consist on defining some targets with their associated trajectories using the VTS Surveillance Simulator. These moving targets are susceptible to be monitored by PTZ cameras depending on its current location. The MAS should perceive sensors information to decide in real-time the best track for its monitoring. As we want to compare the performance between using data fusion or not, the proposed tests will handle different runs with different sources of information, that can be summarized as follow:

- Cameras: In this step each PTZ Agent will only perceive information from its associated cameras. In this way, it can only start monitoring a target if it is within its camera's field of view.
- Cameras+Sensors: This is a second step where each PTZ Agent not only perceives information from cameras, but also information from all sensors deployed in the environment: two radars and one AIS station. Now the agent does not requires to be looking at a given target to know it is present in the environment. Each target will be reported by each sensor covering its location.
- Fusion integrating Cameras+Sensors: In this latter step, the fusion system is represented as an agent. This Fusion Agent will receive all sensors measurements to provide fused target representations to other agents.

C. Cameras Redundancy Test

The first test consists on creating a unique target with a trajectory crossing between all sensors and cameras. As there are more cameras than targets, it is possible that cameras become redundant when monitoring targets instead of panning within its associated region. So the goal here is to understand how data fusion reduces monitoring redundancy when it becomes available.



(a) Multi-Agent System performance using only cameras as sources of information.

(b) Multi-Agent System performance using cameras and sensors as sources of information.

(c) Multi-Agent System performance using data fusion as source of information.

Fig. 4: Set of experiments representing the overall performance of a Multi-Agent System that is coordinating cameras with different sources of information for one target. It can be noticed how the Multi-Agent System is performing better with fusion system as redundancy and untracking values are minimal.

The trajectory described by the vessel starts in $43.4357°, -3.8029°$ (near to Radar1 in figure 3) and ends in $43.4709°, -3.7580°$ (near to Radar2 in figure 3) and travels at a constant speed of $10m/s$ with little maneuvers at $2°/s$. The total trajectory duration is about 580 seconds and it can be completely covered by cameras.

In those experiments are monitored the following states of PTZ Agents:

- Panning Time: Represents the total amount of time where agents have been working in panning mode looking for new targets in their predefined monitoring paths.
- Monitoring Time: Indicates the total amount of time used by agents in the monitoring targets state.
- Effective Monitoring Time: This term refers to total amount of time that agents have been monitoring different targets (non redundant tracks).
- Redundant Monitoring Time: Represents the total time spent by agents for monitoring the same target. This time should be near to zero, as this time should be spent in monitoring non redundant targets (if any) or panning in the environment.
- Lost Time: This term refers to the total time the agents are spending its time in monitoring the environment, or redundant tracks, when there are non-redundant tracks available for monitoring.

The results shown in figure 4 presents the overall performance of the MAS system by using different sources of information.

Figure 4a shows that when using only cameras as a source of information, there are possibilities of targets not being monitored (as there is lost time), as target detection only depends on the current camera's field of view. If the camera is not looking around the target, it gets unnoticed. Also it appears tracking redundancy as each agent will perceive each target as

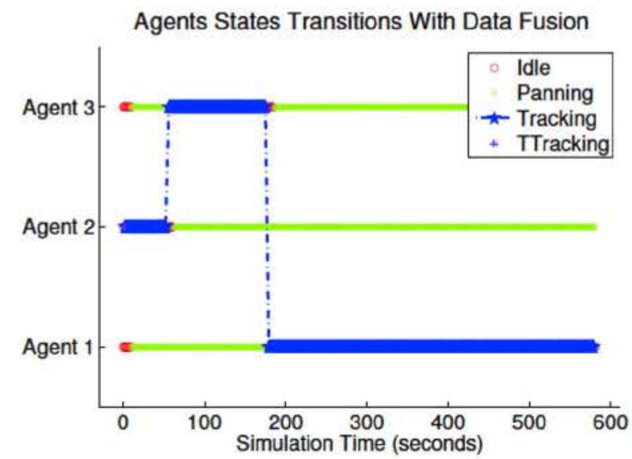


Fig. 5: Agent states when using data fusion as source of information. Notice how the tracking state changes between agents as simulation progresses. There is no noticeable redundancy in those states, as expected when there is only one track in the environment.

an independent track from its own camera sensor. So there are not theoretical conflicts to solve, and some agents will start monitor the same targets.

Figure 4b presents in their turn the use of collaborative and non-collaborative sensors (Radar and AIS) along with cameras. In such deploy, the los time value turns to zero as in this case, the agents does not only depends on camera's field of view, so when a new target enters in the sensors coverage it gets immediately available. However as there are more sensors contributing to monitoring the environment, and those sensors has a greater coverage area than using only PTZ cameras, there is more chance to receive redundant tracks in agents. This is clearly visible, as the redundancy tracking term increased substantially. Also the panning time is reduced as agents spend more time monitoring redundant tracks.

Finally in figure 4c it is used a data fusion system as unique source of information combining all sensors information. In such scenario it is expected to obtain similar lost time values than using sensors, but also reduce the redundancy values as data fusion will provide a set of non-redundant tracks that will allow the MAS applying its conflict solving mechanism as described in III-A. Both redundant monitoring and lost time values in this experiment are minimal as shown in figure 4c, and also panning time is increased. For illustrating how redundancy is minimal we can observe figure 5. In this figure it is shown how each PTZ Agent had a chance of monitoring the track, but none of them overlaps in the same state for the same track.

D. Cameras Occupancy Test

As opposite to the previous test, in this experiment there are enough targets for keep all the cameras in a non-redundant monitoring state (each one looking for a different target). There are two tracks more included to the previous experiment, achieving different trajectories crossing with the original one and similar dynamics.

In this case it is interesting to see how the effective monitoring time is increased when using data fusion, as the redundancy time is nonexistent, as shown in the previous experiment. The panning time in such cases should be reduced as there are enough targets to cover.

It is noticeable how using only cameras (fig 6a), the lost time increases in this test, as there are more available targets, but PTZ Agents are not able to easily find them within its monitoring paths, so they keep panning. However when using also sensors, this panning time has been reduced (figure 6b) for increasing monitoring time. The effective monitoring time is still under the monitoring time as redundancy time is high, similar to the previous test. Notice how in this test with three tracks, the redundancy affects to the lost time, since this redundancy time could be used for monitoring free available targets, which was not possible in the previous experiment that only contained one track.

Finally, the use of fused data clearly takes advantage from using raw sensors, as the effective monitoring time is bigger, and there is no redundancy. However it still appears some lost time, that means that some PTZ agents should be monitoring targets, but lost their time panning. In this case this issue is related to both the MCDM process selected in this paper and the custom targets created in this experiment. For example: in a given moment Agent2 is able to monitor Track1, and Agent3 is able to monitor Track1 and Track2 (Track2 is too far away from Agent2). Both Agents selected to monitor the same Track1, but Agent3 win the bid as described in section III-A. Agent3 will start monitoring Track1, and Agent2 should start panning as it does not have alternatives. This is lost time, as Agent3 should have chosen the suboptimal target Track2, letting Agent2 monitor Track1, so both targets were covered. Thereby, this experiment also let us know that the conflict solving mechanism can be improved taking into account this new considerations.

Also it is clearly visible in figures 6d, 6e, and 6f how the monitored area when using data fusion as source of information is clearly bigger than using cameras or raw sensors. Notice also how the use of sensors (figure 6e) and its monitoring redundancy lead this deploy to decrease monitored areas versus using only cameras or data fusion.

V. Conclusion

In this paper we have extended some previous MAS architecture for automatically controlling PTZ cameras by including different sensors present in common maritime surveillance scenarios, like Radar and AIS. It is presented an VTS Simulation Tool which helped in evaluating the performance of including this new sensors along with a data fusion system. The experiments demonstrates that using a data fusion system is a clear advantage for the overall MAS performance. In this way, the data fusion system is integrated as a new source sensor, which allows agents collaborate for avoid redundancy when monitoring targets, as stated in experiments.

There are still some interesting areas to cover that are marked as future works. The most immediate work is improving the conflict solving mechanism between agents to address the lost time shown in the experiments when using data fusion. Also there is margin to consider different criteria or weights in the decision-making process and evaluate its impact. More experiments can be done by increasing targets density, and including priority zones. Also the MAS is still able of using camera's zoom for getting more accurate target information.

Acknowledgment

This work was supported in part by Projects MINECO TEC2012-37832-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485)

References

[1] N. Park, S. Cho, B.-D. Kim, B. Lee, and D. Won, "Security enhancement of user authentication scheme using ivef in vessel traffic service system," in *Computer Science and its Applications*, pp. 699–705, Springer, 2012.

[2] H. B. Mitchell, *Multi-sensor data fusion*. Springer, 2007.

[3] J. García, J. L. Guerrero Madrid, Á. Luis Bustamante, and J. M. Molina, "Robust sensor fusion in real maritime surveillance scenarios," 2010.

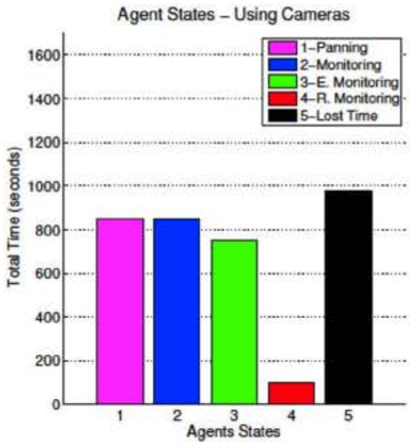
[4] M. N. Murphy, *Contemporary piracy and maritime terrorism: the threat to international security*. Routledge, 2013.

[5] F. Systems, "Flir systems thermal imaging cameras for coastal surveillance applications." Press Release, May 2009. http://www.flir.com/uploadedfiles/cs_emea/application_stories/media/downloads/coastalsurv_en.pdf.

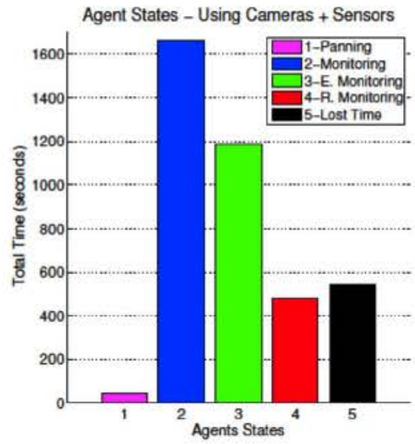
[6] A. L. Bustamante, J. M. Molina, and M. A. Patricio, "Autonomous active-camera control architecture based on multi-agent systems for surveillance scenarios," *Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems*, vol. 109, p. 210, 2013.

[7] A. Luis Bustamante, J. M. Molina, and M. A. Patricio, "A practical approach for active camera coordination based on a fusion-driven multi-agent system," *International Journal of Systems Science*, vol. 45, no. 4, pp. 741–755, 2014.

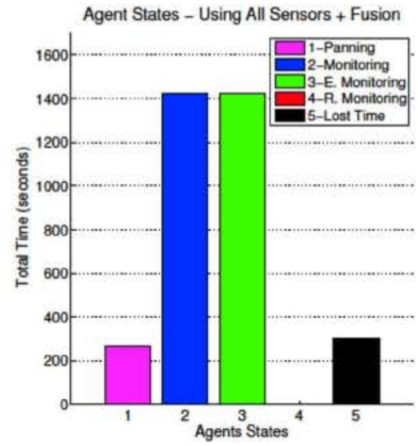
[8] D. D. Doyle, A. L. Jennings, and J. T. Black, "Optical flow background subtraction for real-time ptz camera object tracking," in *Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International*, pp. 866–871, IEEE, 2013.



(a) Multi-Agent System performance using only cameras as sources of information.



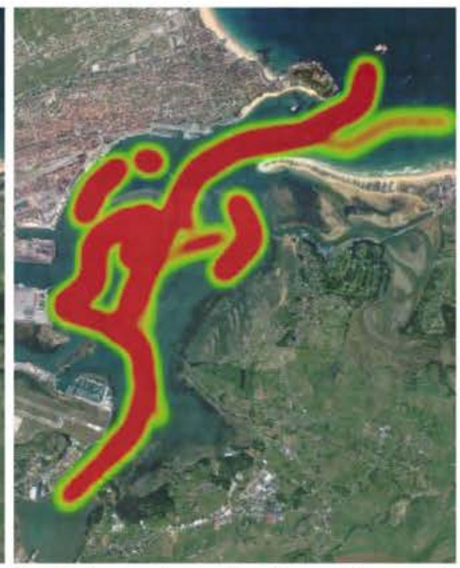
(b) Multi-Agent System performance using cameras and sensors as sources of information.



(c) Multi-Agent System performance using data fusion as source of information.

(d) Heatmap of monitored areas by agents using cameras.

(e) Heatmap of monitored areas by agents using cameras and sensors.



(f) Heatmap of monitored areas by agents using data fusion.

Fig. 6: Set of experiments representing the overall performance of a Multi-Agent System that is coordinating cameras with different sources of information for monitoring three different targets. Also it is presented a heatmap of the monitored areas, and it is clearly noticeable how the area is bigger when using data fusion as source of information.

[9] S. Singh, A. K. Saini, R. Saini, A. Mandal, C. Shekhar, and A. Vohra, "Real-time video acquisition and ptz camera movement control for fpga based automated video surveillance system.," *International Journal of Research & Reviews in Computer Science*, vol. 3, no. 2, 2012.

[10] M. Boussard, M. Bouzid, and A.-I. Mouaddib, "Multi-criteria decision making for local coordination in multi-agent systems," in *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, vol. 2, pp. 87–90, IEEE, 2007.

[11] B. An, K. M. Sim, C. Y. Miao, and Z. Q. Shen, "Decision making of negotiation agents using markov chains," *Multiagent and Grid Systems*, vol. 4, no. 1, pp. 5–23, 2008.

[12] T. L. Saaty, "How to make a decision: the analytic hierarchy process," *European journal of operational research*, vol. 48, no. 1, pp. 9–26, 1990.

[13] T. L. Saaty, "Decision-making with the ahp: Why is the principal eigenvector necessary," *European journal of operational research*, vol. 145, no. 1, pp. 85–91, 2003.