Fusion of Surveillance Information for Visual Sensor Networks*

Óscar Pérez
Computer Science
Department-GIAA
Universidad Carlos III de Madrid
Colmenarejo, Spain
oscar.perez.concha.uc3m.es

Miguel A. Patricio
Computer Science
Department-GIAA
Universidad Carlos III de Madrid
Colmenarejo, Spain
mpatrici@inf.uc3m.es

Jesus García
Computer Science
Department-GIAA
Universidad Carlos III de Madrid
Colmenarejo, Spain
jgherrer@inf.uc3m.es

Javier Carbo
Computer Science
Department-GIAA
Universidad Carlos III de Madrid
Colmenarejo, Spain
jcarbo@inf.uc3m.es

Jose M. Molina
Computer Science
Department-GIAA
Universidad Carlos III de Madrid
Colmenarejo, Spain
molina@ia.uc3m.es

Abstract - The growing interest in surveillance in public, military and commercial scenarios is increasing the need to create intelligent or automated distributed visual surveillance systems. Many applications based on distributed resources use the software agent paradigm. In this work, a multi-agent framework is applied to coordinate an indoor-surveillance system based on video cameras. The capacity of coordination will allow the improvement of the global image and the effectiveness of the task distribution. Software agents are embedded in each camera and control the capture parameters. The multi-agent framework allows the coordination of the acquisition procedure based on high level messages and the fusion of information among agents. The agent paradigm uses the internal interpretation of the situation from each agent to improve the global coordination.

Keywords: Software Agents, Coordination, Data fusion, Active Vision, Sensor Management.

1 Introduction

Nowadays, surveillance camera systems are applied in transport applications [1, 2], public places [3], and etcetera. Research in university groups tends to improve image processing tasks by generating more accurate and robust algorithms in object detection and recognition [4], tracking [5], human activity recognition [6], database [7] and tracking performance evaluation tools [8].

The Third Generation Surveillance System [9] or Visual Sensor Networks are the terms used in the literature to refer to systems with a large number of cameras, resources geographically spread and many monitoring points. Furthermore, these systems mirror the hierarchical and distributed nature of the human process of surveillance. From an image processing point of view, they are based on the distribution of processing capacities over the network and the use of embedded signal processing devices to give the advantages of scalability and robustness of distributed systems. Thus, a distributed solution is an alternative to the coordination problem of multi-camera systems, which is advantageous with respect to centralization in relation to scalability and fault-tolerance. In addition, the distributed nature of this type of surveillance systems supports the proactivity and social skills of camera-agents. Sociability of agents presumes some kind of communication between them. The most accepted agent communication schemes are those based in Speech-Act Theory (i.e. KQML and FIPA-ACL) [10].

The intelligence produced by the symbolic internal model of agents is based on a deliberation about the state of the outside world (and its past evolution), and the actions that may take place in the future. Since our domain cameras need some deliberation about past observations and some predictive ability to improve their future behaviour, then the use of a memory and a certain level of planning should be suggested. Therefore the agents for distributed surveillance systems must be intelligent, and in other words, they should make use of a symbolic internal model. Several architectures can be applied in order to build agents with such deliberation ability. Most of them are based on theories for describing behaviour of individuals, i.e., the Belief-Desire-Intention (BDI) model, the theory of Agent Oriented Programming [11], the Unified Theories of Cognition [12] and the subsumption theory [13]. Each of these theories has its own strengths and weaknesses and supports certain kinds of application domains especially well.

The purpose of this study was the development of a novel framework for deliberative camera-agents forming a visual sensor network. Moreover, the agents exchange information to make a fusion of data in order to improve the surveillance and coordination tasks. In our approach, the distribution is obtained from a multi-agent system, where each camera is represented and managed by an individual software agent. Each agent knows only part of the total information (partial knowledge), and has to take decisions with this limitation. In addition, we have chosen the BDI architecture to implement the deliberation architecture due to its simplicity and physiological background [14, 15]. The well-known software

* Funded by projects CICYT TSI2005-07344-C01-02 and CAM MADRINET S-0505/TIC/0255
This paper describes our multi-agent framework applied to visual sensor networks in distributed surveillance systems. Section 2 deals with one of the biggest challenges faced by researchers in this field. This is the management problem of the information in a sensor visual network. Section 3 describes our multi-agent framework in terms of its reasoning (Belief-Desire-Intention) and perception components (visual sensors). Then, an application of this framework of two-visual sensor network for detecting and tracking intruders in restricted indoor areas is presented in Section 4. We here show the improvements obtained by using this framework analysing the gain with respect to situation where there is no coordination between visual sensors. Finally, the conclusions are given in Section 6.

2 The information fusion levels for distributed vision

A Visual Sensor Network consists of a large number of interconnected cameras equipped with enough processing power to support local image analysis. The extraction of useful information from these networks presents a difficult task due to the huge size of the data-streams captured and the computational demands of the analysis. In previous work, the authors developed several algorithms that processed the information supplied by a unique camera [17-19]. Furthermore, Evolutionary computation was used to obtain the better global performance for various scenarios [20].

In a visual sensor network, each camera is associated to a process that acquires estimates and interprets its local scene. This process is partially based on a tracking system, where the detected moving objects are processed to be recognized and represented so that the location, size and cinematic state of the moving target is always known [18] (see Figure 1).

Figure 1: Structure of video surveillance system

Each node of the system captures the frames of the video sequence and uses them to compute the background estimation. The background statistics are used to detect the contrasting pixels corresponding to the moving objects. These connected pixels must be joined later to form image regions called blobs. Due to some image irregularities, shadows, occlusions, etc., a first problem of imperfect image segmentation appears, resulting in multiple blobs generated for a single target. Thus, the blobs must be re-connected before the track assignment and updating. We use a morphological filtering for improving the segmentation results. Then, the blobs are defined by their spatial borders (usually a rectangular box), area and centroid of the area. The tracker re-connects these blobs to extract all the targets from the background by means of an association algorithm, which analyses the track-to-blob correspondence [17]. This is the

A recursive filter updates the centroid of the area, the rectangle bounds and the velocity for each track from the sequence of assigned values, by means of a decoupled Kalman filter for each Cartesian coordinate. The association process assigns one or several blobs to each track, while the not associated blobs are used to initiate new tracks. The map information and masks are used to tune specific aspects such as detection, track initiation, update parameters, etc. The authors have successfully applied this processing architecture to several video surveillance systems [17, 18]. To illustrate the process, the Figure 2 depicts the different levels of information handled in the system stages (marked in Figure 1 with letters A-D), from the raw images until the tracks.

Figure 2: Information levels in the processing chain

However, we have to integrate all the information supplied by the single cameras within a sensor visual network, and, this is likely the biggest main challenge faced by researchers in this field, i.e., the management of the information flow within the network.

Two main fusion tasks, which are divided into two levels, are identified in this information management for the distributed surveillance system. The first one is a data fusion problem or low-level fusion task which must be solved by collecting and combining all information provided by each individual local processor (the output of the tracking process outlined above). The tracks of the same target simultaneously received by different camera sensors must be combined, being desirable that the information resulting from multiple sources is in some sense better than that obtained if any of these sources were individually used. Every interest object tracked by a camera has information about its localization related to a common reference system. The reference system is
shared among neighbouring cameras, so the association problem of combine tracks from the same target is solved. The result of this fusion process will be used for displaying the results to the human operator, detecting interesting events, etc., and should take advantage (in terms of robustness, accuracy and comprehensibility) of the data fusion output. The high level fusion task is the coordination between nodes: each camera will take decisions that must be coherent to have a global and effective network operation. The problem of distributing a task over several cameras, according to their individual ability to contribute their local data to a desired global solution, is particularly complex for large-scale camera networks. Assuming that a number of cameras can solve the same recognition and tracking tasks, the point is how to choose the most appropriate camera for each task. This can be regarded as a coordination problem in which the goal is to maximize the reliability of the managed information by each node of the surveillance system.

A centralized decision process could be theoretically considered, applying the information generated in the fusion process to take all the individual decisions corresponding to all nodes. This approach would have scalability problems to consider the detailed situation of all nodes in a big network, and would not take advantage of the processing capabilities distributed in the network. Conversely, here we present a novel framework for deliberative camera-agents system design in a visual sensor networks, taking into account this problem of management of the information flow in the network. Each agent deliberatively takes decisions to carry out the system tasks coherently with other agents, considering both the information generated in its local process and the information available in the network. In this coordination process, the situations of transitions among areas covered by different agents will be the most relevant. An example will be described and used for evaluation in the following sections.

3 Multi-Agent Framework

In this section we describe the use of a BDI (Beliefs-Decisions-Intentions) model on agents to coordinate cameras in a surveillance domain. The final goal of agents is to improve recognition and interpretation process (object class, size, location, object cinematic) of mobile targets through cooperation, and therefore, to improve the surveillance performance of the whole deployed camera system. The cooperation takes the form of messages between agents that contributes to a better performance of a distributed surveillance system giving specific information about mobile targets in critical situations (such as when such targets go out of the scope).

In order to act rationally, BDI model represents internally the situation faced and the mental state in form of beliefs, desires and intentions [14, 21-23]. We mean as Beliefs of an agent its own internal state and its assumptions about the current state of the world according to what it perceived from outside. The Desires capture the motivation of agents; they specify preferences over future states of the world, and rule the course of decisions. Intentions are the basic steps which an agent is committed with, in order to satisfy its desires. The sequence of them (and of subgoals) forms plans, which also play a relevant role in BDI paradigm. Plans are the means by which agents achieve their goals and react to outside events.

Considering agent architecture and coordination aspects, our multi-agent framework for building distributed surveillance systems has its own beliefs-desires-intentions. Beliefs of a camera agent represent information about the outside world as mobile objects that are recognizing and tracking, the other camera-agents with which it can cooperate, and physical objects that can be relevant such as entrace-points (doors, windows) and obstacles that could avoid a right vision (tables, closets). The final goal of each camera-agent is the right recognition and tracking of moving objects (assumed to be intruders). So camera-agents have two desires: permanent surveillance and temporal tracking. And finally, we can basically distinguish between external and internal intentions. External actions correspond to communication acts with other camera agents, while internal actions involve recognition and tracking tasks carried out by each agent.

3.1 Camera Agent Plans

The plans related to the agent-camera desires are depicted in Figure 3. The agent surveillance plan (invoke from initial) interprets permanently images from the camera. When a potential moving object is detected (probably an intrusion is suddenly recognized), a tracking subplan will then be initiated with the purpose of tracking such moving object.

Additionally, the internal process of recognition and tracking (ruled by the tracking plan), may produce internal events due to two possibilities:

a) the tracked moving object is close to a zone of limited vision (for instance doors and windows), and it is expected that in the near future the moving object will be out of the scope of the camera.
b) the moving object it is already out of the scope of the camera.

In the first case, the agent will warn about the expected appearance of the moving object to the agents that rule the closest camera agents starting a call for proposals dialogue that is performed by another subgoal: warning.

In the second case, the agent queries other agents that could possibly view such disappeared moving agent, in order to determine if the moving agent definitively left the scope of the camera or not (and therefore whether the internal tracking process should be terminated or not). The implementation of the query dialogue is performed by another subgoal: querying.
Finally camera agents require another plan to confirm/reject the presence of a given moving object when another agent queried for it. This plan just evaluates whether the moving object is visible from its camera or not, and then inform to the other agent of the result of such evaluation. This type of plans is especially relevant to deal with transitions from areas covered by different agents and guarantee a successful hand-over.

This hierarchical structure of plans in the agents’ desires can be implemented with JADEX framework. For instance, the tracking plan generated after the perception succeeds in detecting an interesting object (such as an intruder) is instantiated as a subplan, with the corresponding xml lines for the plan and subgoal:

```xml
<plan name="surveillanceplan" instant="true">
  <constructor>new Surveillance()</constructor>
  <performgoal name="trackinggoal">
    <parameter name="intruder" class="Object" />
  </performgoal>
</plan>
```

The case of a warning describing object appearance to be sent to agents in charge of close cameras is also instantiated as a subgoal: warning:

```xml
<plan name="warningplan">
  <constructor>new Warning($event.goal.intruder)</constructor>
  <filter>new jadex.GoalEventFilter("warninggoal")</filter>
  <performgoal name="warninggoal">
    <parameter name="intruder" class="Object" />
  </performgoal>
</plan>
```

The queries to other agents about disappeared objects are instantiated as other type of subgoals: querying:

```xml
<plan name="queryingplan">
  <constructor>new Querying($event.goal.intruder)</constructor>
  <filter>new jadex.GoalEventFilter("queryinggoal")</filter>
  <performgoal name="queryinggoal">
    <parameter name="intruder" class="Object" />
  </performgoal>
</plan>
```

3.2 Camera Agent Coordination Plan

Messages interchanged among deliberative agents fit the FIPA standard that adds a performative to each communicative act. For instance, the corresponding FIPA performative for “Query” dialogue is ‘query-if’. The agent performing the query-if act has no knowledge of the truth value of the proposition (moving object location), and believes that the other agent can inform the querying agent about it. So the receiver agent would answer with an ‘inform’ FIPA communicative act:

```xml
{query-if
  :sender (agent ?j)
  :receiver (agent ?i)
  :content (visible (object ?x))
  :reply-with queryx
}
```

```xml
{inform
  :sender (agent ?i)
  :receiver (agent ?j)
  :content (not (visible (object ?x)))
  :in-reply-to queryx
}
```

Broadly three main types of messages can take place between the agents for the distributed-surveillance application. The first of them (“Call for proposals”) intends to warn the receiver agent about the expected future presence of a moving object, in order to produce a tracking process of the receiver agent over such moving object. This warning takes the form of a proposal. The second one (“Inform”) looks forward a confirmation of the presence of moving objects in the scope of the receiver agent. This communication takes place when expected circumstances were accomplished (for instance, when moving objects leave one room and enter into other one). But it can also take place standalone if such circumstances were not expected, and occur suddenly (the moving agent disappears from the scope of a camera agent but no door/window opening was predicted/observed (for instance the moving agent may be hidden behind a closet or table). So, the third type of communication, “Query”, intends to obtain a confirmation of that other agent is viewing such moving object. These types of messages, depicted in figure 6, can be basically seen as information interchanges, with little possibility for negotiation or refinement. Agents receive this information and, considering also their local information, take the decision about the appropriate plan to activate.
3.3 Experimental Scenarios

In previous works, the surveillance system was tested for different scenarios, both in outdoors (pedestrians in walking footpaths, traffic in a road, aircraft in parking areas) and indoors areas (people walking in corridors and rooms).

In order to evaluate the performance of our multi-agent framework in coordination tasks, we have applied it to solve different problems. As an example, we chose the indoor example or the "laboratory surveillance" situation. The application is an indoor application in which two agent-cameras detect intruders in a restricted room. The first agent controls the corridor where the room is. Once it has detected an intruder and checks that this one is close to the door that access the room, the corridor agent sends a message to the agent-camera inside the room so that this second agent is alert. The message contains not only the warning to inform about an intruder but also the information of this intruder: size, cinematic, etc. That results very useful for the room agent because the restricted room has many objects that may occlude the stranger and the lights might deform the person and confuse the agent. So, the main dialogue between agents uses the warning dialogue and query dialogue. With this scenario, we demonstrate that our framework that fusion key information of two agents is more reliable and robust than the one without agent coordination.

Figure 5: Sample scenario used in this section

For this scenario containing two BDI agents, we have taken several premises:
- There is only one intruder,
- The intruder moves from the corridor to the room through one of the doors that link both of them.
- No commands (focus, turnover) to the camera are required to implement the agent tracking.
- Each camera may observe all the room or corridor.

Let us assume that an intruder try to go into the room from the corridor through the door2. The agent camera of camera2 must warn ("Call for proposals") to the agent camera of camera1 about this fact. In the message, the agent camera of camera2 specifies the intruder location, size and identification. Thus, the camera agent of camera1 is able to initiate a tracking plan with the supplied information of the other camera agent. In order to assess the improvement of the global surveillance system by means of this multi-agent framework we will show the obtained surveillance results of two experiments. We have evaluated the tracking plan performance of the agent camera 1 with and without the supplied information from the agent camera 2. Next subsection, we explain the results of the two experiments.

4 Experimental Evaluations

Concerning the “laboratory surveillance” scenario described in previous section, we are going to evaluate the improvement of the surveillance system by means of agent coordination, compared with the isolated operation of a particular node. An agent surveillance plan is able to follow all kinds of targets and their different movements all around the camera plane. The effect of using the flow information coming from its neighbour agents should increase the reliability of estimations in the agent, as it will be assessed along this section.

4.1 Evaluation System

With the purpose of evaluating the tracking system, we suppose that the intruder enters the room and moves following the wall from door2 to door1. This trajectory is used as ground truth exclusively to assess the system performance under these conditions (it is not information available in the agent). We have selected 15 samples of this intruder action which we have evaluated with and without information exchange between both agents. The quality measures of both experiments were computed considering as an input data the 15 video sequences and the path followed by the intruder.

The authors have previously applied evaluation metrics to assess video surveillance systems [19]. In our evaluation system, each time a track is initiated or updated, the agent tracking plan calls to the evaluation system. In order to have a more detailed idea of the system performance, the agent camera plane is divided into 10 zones (see Figure 6). Each zone is defined as a fixed number of pixels of the x-axis, the 10% of the horizontal size of the image. The horizontal component has been selected to analyze the metrics because it constitutes the main coordinate in which the objects move in this particular study.

Figure 6: Segmentation of each frame in 10 zones

The metrics that we have applied to both experiments are:
- Initialization: It is the number of frame where the intruder is detected by the agent tracking plan.
- Absolute Area Error: It is computed by calculating the area of the detected track. It is important to measure the absolute area in order to have an idea of what the camera is really tracking. For example, in this case, the lights of the room make the intruder look bigger than her real size. Thus, the cameras without coordination track not only the shape of the person but her shadows. The fusion of information between agents avoids this problem by adapting the track to the real size.
- Transversal Error (\(d(P,r)\)): It is defined as the distance between the center of the bounding rectangle (P) and the segment (r) which is considered as ground truth.
- Inter-frame Area Variation: This metric is defined as the variation of area between the current and the previous update of the track under study. It is required to check that the previous track exists. Otherwise, the value of this metric is zero.
- Continuity Faults: The continuity faults metric is only measured inside a gate defined by the user. This gate is chosen in such a way that represents the area where no new tracks can be appeared or disappeared because the intruder has already turned up on the right side of the image. This number checks if a current track inside the gate existed the previous moment or did not. If the track did not exit, it means that this track was lost by the agent tracking plan and recovered in a subsequent frame. This behaviour must be computed as continuity fault. This continuity metric is counter where one unit is added each time a continuity fault occurs.
- Number of Tracked Objects: It is known that there is only one intruder per video, but the agent tracking plan may fail and keeping more than one or zero in some situations. Thus, each time a track is initiated, the agent surveillance plan marks it with a unique identifier. This metric consists of a counter which is increased by one unit every time that a new object with a new identifier appears in the area under study. After the evaluation of the whole set of videos, this metric is normalized by the total number.

Later on, some frames later, the initialization is confirmed when the person enters the room. Otherwise, the initialized track must be removed. In contrast, if the tracking system has no previous knowledge, the initialization will be carried out after surveillance plan of the agent camera detects the intruder.

Table 1: Comparison of the initialization of an intruder track for the two available systems. The system with agent architecture initializes the track when a message from the extern camera is received by the inner camera (frame number 1).

<table>
<thead>
<tr>
<th>Number of videos</th>
<th>System without agent architecture</th>
<th>System with agent architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>Initialization frame</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>1</td>
</tr>
</tbody>
</table>

Second of all, the absolute area of the tracked object with the agent coordination activated is almost constant as it can be seen in Figure 10 (right-hand side), compared with the isolated case (left-hand side). We can observe that the area on the right-hand side has a very lower variation and keeps almost constant compared with the situation on the left-hand side. The graphs in Figures 10 and 13 have a continuous line with the mean value, two segmented lines around the continuous one with the standard deviation (+−q) and two dotted lines with the maximum and minimum values. The graphics are divided horizontally in 10 zones representing the whole area covered by the agent surveillance plan.

The effect on the estimated area is because the agent camera of the corridor sends stable information about the location and size of the intruder to the agent camera of the room. This agent performs a fast initialization and rebuilds the representation, which is updated later with the observation generated by the own camera. Thus, the surveillance system processes some blobs which are completed with the knowledge passed in the message: height and width of the person. Therefore, the surveillance system tracks the available blobs (some of them are impossible to detect due to occlusions) and reconstructs the original size. Furthermore, this computation allows us to avoid taking into account shadows and reflections because this spurious information tracked by the surveillance system will not fit with the previous information and will be discarded. The Figure 11 shows us the points marked as pixels in motion. Many of them are spurious information due to the light coming into the room when the door is opened and the reflection of this light in the wall. The system is able to reconstruct the position and the size of the intruder and removing the false information.

4.2 Performance Results

The following tables and graphics compare the tracking system performance without and with the agents’ coordination operating in the system.

First of all, we can observe in Table 1 that the system inside the room initializes the intruder track as soon as a message with the information of the intruder is available.
Thereby, the inter-frame area variation or the variation of the area between the current and the previous update of the track under study of our new system is very low.

The next pictures give us a clear idea of the system performance. The Figure 10a and 10b are two frames of a video sequence, Figure 10c and 10d show the points marked as pixels in motion and Figure 10e and 10f contain the output of the system. Thus, the Figure 10c shows the blobs processed by the system for the Figure 10a. The system can not capture more blobs of the intruder as there are some obstacles (tables and computers). The surveillance system gives as a result the intruder track that is depicted in Figure 10e by the smaller rectangle. Nevertheless, the agent coordination rebuilds the intruder track by using the previous knowledge of the intruder’s size. The same process is showed for the second Figure 10b. In this last case, the obstacles allow the surveillance system to capture more pixels so that the system rebuilds less parts of the intruder.

As we can observe in the last images of Figure 10, the inter-frame area variation of the rebuilt track is low due to the reconstruction of the track that allow the system to avoid shadows, reflections and rebuild the original size of the intruder.

In the figure 11, the transversal error with respect to ground truth is depicted for both cases. We can see that the error is almost zero for the second architecture (right-hand side), due to the adjustment of the track to the previous knowledge. As we said before, the system takes the track given by the surveillance system and rebuilds it by using the characteristics of the intruder. In both cases, the system consider as ground truth the line defined by the mass centres of the complete person, that is, the centres of the reconstructed tracks from the door 2 to the door number 1.

Finally, the number of tracked objects shows that the system with agent coordination keeps a right representation (an intruder) in zones 8, 9 and 10, which are the areas close to the door number 2, and performs a smooth transition to the own detections (from zone 7 to the left). That is because the system initializes the intruder track from the very beginning, while the system without agent coordination presents a considerable delay in this initialization.

FIGURE 8: Absolute area for the architecture without (left picture) and with (right picture) agent coordination

FIGURE 9: Reconstruction of the track based on the previous knowledge

FIGURE 10: System performance

FIGURE 11: Transversal error for the architecture without (left picture) and with (right picture) agent coordination

FIGURE 12: Continuity faults for both architectures
5 Conclusions

In this work a multi-agent framework has been applied to the management of a surveillance system using a visual sensor network. We have described how the use of software agents allows the design of a more robust and decentralized system, so that the management is distributed between the different camera agents. The architecture of each agent and his level of reasoning have been presented, as well as the rules of coordination. We have shown the improvement of using this framework to coordinate a visual sensor network settle in an indoor surveillance space using several numeric performance metrics. The capacity of software agents to represent real situations has been analyzed, and how the information interchanged improves the coordination among the camera agents.

References


