A Network of Sensors Based Framework for Automated Visual Surveillance

Ruth Aguilar-Ponce, Ashok Kumar, J. Luis Tecpanecatl-Xihuitl and Magdy Bayoumi
Center for Advanced Computer Studies
University of Louisiana at Lafayette
PO Box 4330, Lafayette, LA USA 70504-4330
ak@cacs.louisiana.edu

Abstract

This paper presents an architecture for sensor based, distributed, automated scene surveillance. The goal of the work is to employ wireless visual sensors, scattered in an area, for detection and tracking of objects of interest and their movements through application of agents. The architecture consists of several units known as Object Processing Units that are wirelessly connected in a cluster fashion. Cluster-heads communicate with the Scene Processing Units which are responsible for analyzing all the information sent by the former. Object detection and tracking is performed by cooperative agents, named as Region and Object Agents. The area under surveillance is divided into several sub-areas. One camera is assigned to each sub-area. A Region Agent is responsible for monitoring a given sub-area. First, a background subtraction is performed on the scene taken by the camera. Then, a computed foreground mask is passed to the Region Agent, which is responsible for creating Object Agents dedicated to tracking detected objects. Object detection and tracking is done automatically and is performed on the Object Processing unit. The tracking information and foreground mask are sent to a Scene Processing Unit that analyzes this information and determines if a threat pattern is present at the scene and performs appropriate action

***Key words*: cooperative agents, sensor network, object detection, image understanding**

1. Introduction and Related Work

The changing world scenario where the security is at risk for several factors including industrial espionage and terrorism requires sophisticated surveillance. The type of surveillance systems can provide information with details such as the date, time, and persons of interest. For example, in an airport, surveillance system can be used with a database of known persons of interest for detecting and tracking them to monitor their activities. The demand for automated system is increasing for and state–of–the–art video surveillance systems are becoming more sophisticated through research. They typically consist of several cameras distributed through an area connected to monitors in a central operator room, where highly qualified personnel are in charge of reviewing and analyzing the video stream of each camera to observe suspicious activity. In other words, the technological advances pertain to networking and computing capabilities while there are challenges to overcome before a reliable automated surveillance system is realized [1]. These technical challenges include system design and configuration, architecture design, object identification, tracking and analysis, restrictions on network bandwidth, physical placement of cameras, installation cost, privacy concerns, and robustness to change of weather and lighting conditions.

In this work a distributed sensor network system is proposed. A distributed sensor network is a large collection of sensors distributed through an area of interest where each sensor possesses a processing capability. The sensors cooperate in an ad-hoc manner to communicate information to each other. A sensor collects data and performs basic processing on the collected data. The information of interest is a general view of the monitoring area rather than a single view of one sensor. Therefore, the information collected from all parts of the network is merged by means of data fusion. The resulting data is then used to infer the state of the monitoring area. The increasing interest in sensor network comes from the variety of applications that such networks have the potential to effectively address. There are several applications that can benefits from wireless sensor network such as surveillance of battlefields, intrusion detection, improvement of intelligibility under noisy conditions, automatic control of air-conditioners and humidifiers in buildings [2].

The task of a distributed sensor network is to obtain data from the sensors that are spatially dispersed and process it, eliminate noise that could corrupt the signals, integrate the data and reduce the uncertainty in it. Once this process has finished, an interpretation of the data must be performed. As a consequence, the sensor network must comply with different characteristic such as [2]: the network must have intelligence on each node; the performance must not degrade because of the spatial distribution and the network must accommodate diverse sensors. Furthermore, the network must be resilient to node failure. The network must operate even if some nodes have failed. Data processing must be done in real-time, because sensor data are perishable. Since communication will consume most of the node power, a trade-off between communication-computation must be achieved. Several characteristics of distributed sensor network are desirable for an automated surveillance system. If each camera in the area of interest is equipped with intelligence to automatically detect and track moving objects on the scene, more information will be available for detection of specified activities. Surveillance system will also require an abstract interpretation of the scene under investigation and not only have the images captured by the cameras.

Since early 90's, agent paradigms have been subject of intense study. Agents are an extension of the object-oriented paradigm and they have been suggested as a breakthrough of computer science [8]. An agent is a computer system capable of autonomous action in its environment in order to achieve its goal. An agent must be capable of controlling its state and behavior as well as interacting with the other agents. The main difference between an agent-based system and an object-oriented system is the autonomy. The object encapsulates states and methods. The methods can modify the state of the object; however the object cannot modify its own state. An agent encapsulates its state and its behavior. While an object invokes a method, agents request actions to be performed. The decision to grant the request lies with the recipient. Agent based system are designed by first recognizing which part of the program can be a candidate for an agent. The entities that are the best candidates must be autonomous, social, responsive and proactive. Autonomous entities do not require human interventions to perform their tasks, while they are socially capable of interacting intelligently with other entities. They must be responsive to their environment and capable of taking initiative, i.e. proactive. The problems that are suitable for multi-agent solutions include problems requiring interoperability and interconnection such as decision support systems [9], and problems whose solutions are drawn from distributed autonomous experts, for example health care providers.

Problems that are inherently distributed like distributed sensor network.

Agents cooperate between them to achieve their own goal. ISAAC has been proposed as a method to analyze team behavior based on agents to improve the overall performance [10]. An effort to support mobile computing environment on database systems is presented in [11]. This architecture employs agents as a representative of mobile system. Agents have also been employed in marketplace modeling. One of the major research topics in these applications is the negotiation protocol. An extended Multi-Agent Negotiation Protocol is present in [12]. Agent-based system improves user's ability to manage information, especially if the information is distributed through a network and is changing daily. A framework for mining information based on agents is presented in [13]. Another popular application of agents is in automation of different tasks. Automated system for multiple sensor placement based on coordination of intelligent agents is presented in [14].

Agent paradigm offers several desirable characteristics such as autonomy to achieve a given goal, cooperative efforts between agents, and responsiveness to the environment as well as taking initiative to perform a given task. Therefore, an automated surveillance system based on agents can perform detection and tracking without human supervision.

1.2 Related Research

There have been several efforts to creating an automated surveillance system. Some of these efforts are summarized in this section.

Video Surveillance and Monitoring (VSAM)

An early research effort to address this problem was a system proposed by Collins, et al. at Carnegie Mellon University - Video Surveillance and Monitoring (VSAM) [15]. VSAM architecture consisted of an Operator Central Unit, several Sensor Processing Unit, Graphical User Interface (GUI), and Visualization nodes.

The Sensor Processing Unit performs object detection by means of adaptive background subtraction and three-frame differencing. This information is sent to the Operator Central Unit. Sending only the information of the detect object saves bandwidth as well as power. The sensor processing unit combines camera and a local computer connected by fiber optic. The Operator Control Unit receives processed data from all the sensor processing units distributed through the area of interest. Once the operator control unit acquires results from the sensor processing unit, this information is integrated with a site model and a database of known objects to infer information of interest to the user. The resulting data is sent to the GUI and visualization nodes. The user must indicate surveillance task that must be performed. The operator control unit works as an arbiter, indicating which unit must perform a given task. The assignment of task is done by a visibility analysis based on the site model and the geographic location of sensors. The GUI provides an interactive interface between the user and the VSAM network. The GUI consists of a map of the area overlaid with sensor locations. The user assigns tasks by means of the GUI. The tasks include tracking of a given object, displaying fields of view or creating a region of interest. Visualization nodes provide several point of data access through the VSAM network.

This system has several strong points such as several points of visualization through the network, saving bandwidth by sending symbolic information about detected object and automated detection. However the system still needs human supervision. Tracking of moving objects is only performed if the user assigns the task. Operator control unit acts as an arbiter, therefore if it fails, tasks cannot be completed. The architecture detects and classifies object present on the scene and shows all this information in a report to the user and expects the user to assign tasks.

Tracking Moving Object System

One of the major tasks of any surveillance system is tracking detected objects present on the scene. A tracking system based on agents has been proposed by Shakshuki, et al. at Acadia University [16]. This approach assumes that the moving object is equipped with a Global Positioning System (GPS) receiver. The region of interest is divided into several circular zones. Each zone is assigned to an agent, which resides in the center of the zone. If the object is inside the agent zone, it will track and record the object information until the object is out of reach. Each zone is subdivided into an inner and outer circle. When an object leaves the inner circle, the agent negotiates with the agent of the next area so that the object can be properly handed off. However, this approach cannot be used in surveillance systems because there is no GPS information available (i.e., a suspicious person or vehicle is not likely to carry a GPS system). Also the agent of each region must track information of all objects inside its zone, but if multiple objects are inside its zone only one agent may not be capable of tracking all objects.

Distributed Intelligence for Visual Surveillance

A framework based on cooperative agents for visual surveillance was proposed by Remagino, et al. at Kingston University [17]. The system consists of several cameras distributed through the area of interest where the field of view of the cameras can be overlapped. The framework defines three agents: Ground Plane Agent, Camera Agent, and OA. The Ground Plane Agent is responsible for calibrating the camera to a common ground plane coordinate system to allow integration across multiple cameras. The camera agent is responsible for detecting and tracking all moving object crossing its field of view. The detection of object is done by classifying each pixel as moving or static compared with the previous frame. Once an object has been detected, the system computes the 3D trajectory of the object and extracts the set of sub-images containing the event on each frame. If multiple agents are tracking the same event, then they must merge and inherit the data channels from multiple camera agents

The detection algorithm employs classification of each pixel as moving or static. However it is not clear whether this detection algorithm will discard shadow cast by a moving objector if the algorithm is robust against changes of illumination. The behavior models are limited to normal activities for persons and vehicle entering or exiting an area. There is no threat analysis. Several cameras may instantiate a number of agents tracking the same event, which may results in waste of resources until the agents merge.

Computer Vision System for Modeling Human Interaction

A real-time computer vision and machine learning system for modeling human behavior and interaction in a visual surveillance system is proposed by Oliver et al. at MIT in [18]. Visual surveillance requires understanding how humans behave and interact among them to achieve a secure area. This approach models person-to-person interaction using statistic learning techniques known as Hidden Markov[19-20] and Coupled Hidden Markov Models [21] to teach the system to recognize normal single-person behaviors and person-to-person interactions. The system employs a camera with wide field of view watching a dynamic outdoor scene. Once an image has been acquired, motion detection is performed by subtracting a background model from the current scene. The background model is built by averaging N previous frames. This technique is known as eigenbackground subtraction. Tracking of each event is performed by a Kalman filter that predicts the event positions and velocity in the next frame. The computer vision module extracts a feature vector for each detected object describing its motion and heading as well as spatial relationship to all nearby moving objects. The hidden Markov models take the feature vector to classify the perceived behavior. The eigenbackground subtraction method used to detect objects requires N previous frames, covariance matrix and mean background image. Therefore, the system will require massive amount of resources to achieve real-time operation. Visual surveillance requires interpretation of image sequences; to achieve this, Bayesian networks have been employed. A decomposition consisting of perceptual processing and conceptual processing of the dynamic object present on the scene is proposed in [22]. A summary of the previous research compared with our proposed architecture is summarized in Table 1.

Our approach proposes a distributed network of several sensors. Each sensor is handled by an Object Processing Unit (OPU), which communicates its finding to a Scene Processing Unit (SPU). While the OPU is in charge of detecting and tracking moving object present at the scene, SPU will analyze the information from all OPUs, in order to detect threat pattern and perform proper action. OPU use an agent-based approach for detecting and tracking objects.

The organization of the rest of the paper is as follows. Section 2 provides the details on the benefits and features of our model. Section 3 describes the proposed architecture and components. Section 4 discusses the design and simulation of the system based on agents, followed by conclusions.

2. Benefits of the Proposed Model

Existing surveillance systems send video captured by Close Circuit Television cameras to a central unit, where several persons look at monitors and decide whether the scene presents a security threat. An automated system will avoid necessity for operators to monitor these scenes.

The system will help to identify persons entering an area under observation and provide tracking information on every person. The followings are the features of the proposed architecture for an automated scene understanding:

- Resistance to Attacks. We propose a distributed sensing and coordination effort so that the surveillance network can be operating even when a part of the network has been disabled through a malicious attack.

- Tracking of Objects by Agents. The cameras distributed on the area of interest may not be sufficient to track all the objects at a given time. Therefore an approach based on agents is applied. Agents are assigned for detected-moving objects on the scene, and help to perform the proper handoff between cameras.

- Automated Detection and Tracking. The OPUs distributed in the area of interest detect objects present on the scene by means of background subtraction. Once an object has been detected, cooperative agents track them automatically. This process eliminates task assignment.

- Merging of OPUs data before sending it to the SPU saves bandwidth as well as transmission energy. Furthermore, the SPU only has the relevant information because fusion process takes care of the redundancy on the data.

- The architecture includes an object tracking system based on multi-agents paradigm. The tracking system includes two agent models called *region* and *object* agents. A cooperation algorithm for tracking events inside the area of interest is also proposed. Teaming between RAs also provides proper event hand off.

3. Proposed Architecture

The proposed automated scene understanding architecture consists of two basic units, Object Processing Unit (OPU) and Scene Processing Unit (SPU). Figure 1 illustrates the relationship between them. Several OPUs form a cluster and one of them is designed as a *cluster-head*. A cluster-head is responsible for integrating all the information coming from the cluster members. Once the data has been merged, it is sent to the SPU for further processing. The OPUs are responsible for detecting moving objects present in the scene, and they share this information among them by means of wireless connections. Each cluster covers a specific area. Therefore, the information in a given cluster is highly correlated and useful as it gives us more information on the moving objects and positively impacts the false alarm ratio.

There are several SPUs distributed throughout the network, providing several points of visualization by means of the Graphic User Interface. SPUs analyze the scene information, from all the cameras, in the cluster connected to it. Object classification into several predefined set is performed into this unit. Threat analysis is performed by a statistical learning technique discussed in section 4. Figure 2 illustrates the connection of several OPU-clusters and scene processing units.

The information that flows through the network is only the object model information, i.e. no video is transmitted from the OPUs to SPU. This way the use of bandwidth is optimal. However, the architecture allows the possibility of viewing the actual video footage on the cameras. The dashed links in Figure 1 from the OPUs to the SPU represent this possibility. On a given time the user can indicate by means of the Graphic User Interface his desire of watching the video of a given camera. The OPU will then suspend processing and compress the video in order to send it to the SPU. Therefore, each OPU has two action modes. The *Default Mode* processing will detect and track object and send only the object model information. The second mode is *Video Mode* in which the OPU sends compressed video to the SPU. This option is available in the case when an expert's intervention is in place or the user needs to take a careful look on a possible threat already detected by the system in order to avoid a false alarm. The GUI reports the actions on the scene and provides a method to review video from the desired camera at a given time.

The presented architecture may be used for surveillance in buildings, streets and parking lots adjacent to the buildings under observation, and on borders etc. The cameras can be deployed indoors and outdoors. The OPUs related to each camera detect object present in its range of view and assign software agents to track each object. This information is sent to a SPU located inside of the building under surveillance. SPU is able to classify moving object, as person, group of persons, automobile etc., which through further processing, are decomposed. Then, all the information is used to identify possible threat and perform adequate action. Figure 3 shows a deployment of a unit outdoors and indoors of a building.

3.1 Object Processing Unit

The OPU is in charge of sensing the area of interest and identifying moving objects in the scene. In order to perform its functions, an OPU will consist of four basic units: sensor, processing, transceiver and power unit.

The OPU obtains the sensor data that could be image, seismic or acoustic, and then performs object detection. Once an object has been detected, the object model parameters are updated using the recent lectures of the sensor. When the sensors have some data to be reported to the SPU, a cluster-head is determined between all sensors in the cluster. The cluster-head is chosen based on the available energy of the nodes in a cluster. The OPU performs pre-processing on its own data, after which the data is sent to the cluster-head. A Cluster-head is in charge of merging this data, so that only the valuable information is sent to the SPU. This way bandwidth and energy is saved. Figure 4 shows a functional diagram for a general OPU.

A video sensor provides more information about moving objects than any other sensor. Using several image frames, we can determine velocity, acceleration and heading of the detected object. Our proposed image sensors' functional diagram is shown in Figure 5. Once the image has been obtained, object detection is performed. The object detection is performed by means of background subtraction. Background subtraction provides a foreground mask where only the moving objects are present. Foreground mask is passed to the cooperative-agent system, which is in charge of obtaining the object model parameters and tracking the present objects on the scene. Tracking systems based on agents are discussed in the next section. The object model information along with the image of the object and model values are sent to the SPU. The system is based on fixed-lens cameras in the current stage of development (i.e., the zoom option is not considered currently).

3.2 Object Detection

There are two well-known techniques for object detection in a video stream: optical flow and background subtraction. Optical flow is capable of detecting object movement even when the background is also moving. However this technique is computationally complex. On the other hand, background subtraction is more suited to detecting movement or changes on the scene, yielding to a low complexity implementation. A fixed background is a requirement for this type of detection. Since we are considering only fixed cameras, background subtraction is a better candidate for our application. In addition, object detection must be performed in real time and using the least amount of memory because the resources of the OPU are limited.

Background subtraction is done in several steps; first, the raw image is changed into a format that can be processed. Then a background model is built based on previous frames. Next, comparison between input frame and background model is performed. As a result, a foreground mask is built. Several methods have been proposed over the years, with the main difference on background modeling. This process can be done recursively or non-recursively. Recursive techniques do not require storage of previous frame but are highly complex, while non-recursive techniques require frame storage but are less complex.

3.3 Scene Processing Unit

Object model parameters and tracking information of these objects are sent to the SPU. Several OPUs can detect the same object, therefore through the data fusion performed at the cluster-head, multiple instantiation of the same object are merged. The SPU has a broader look at the scene under observation because it receives the information from all OPUs in the same area. Thus, the SPU can perform more accurate threat detection. Along with object model and tracking information, a segment of the image containing the detected object is received by the SPU. This segment is used by the classification and decomposition procedure. The object is classified as a person or vehicle. A person set contains a single person or a group of persons, while the vehicle set contains sedan, truck, delivery truck, 18 wheelers. If the detected object has been classified as a person, then decomposition procedure takes place. A person is decomposed into legs, arms and trunk. This decomposition is performed in order to determine if the person has a hazardous material in his arms, or strapped on his trunk, like a weapon, etc. This information is sent to the threat detection procedure. Based on that information the SPU decides what action will be proper, such as taking a closer look on the scene or an actual alarm detonation. Figure 6 shows a functional diagram of the activities performed at the SPU. The tasks performed by the SPU are computationally complex, thus require high computation capabilities and memory as well as a database of threat patterns.

3.4 Graphic User Interface

A GUI provides a daily report of action on the area of interest. The report includes all information collected from OPUs on the particular object observed. But it also provides a tool to observe video from the cameras. The video information will be sent compressed to the SPU, and the SPU will be in charge to decompress the video and show it by means of the GUI.

3.6 Model for Multi-sensor Data Fusion

In a wireless sensor network, the information of interest is a global picture of what is happening inside the area of interest, which is more than an individual scan reading for each sensor. Therefore, a method to integrate all the data from each sensor must be provided.

The model presented in [23] takes the data from multiple sensors and sorts them into the different task assigned to the system. Then, the data is transformed into frame to identify and classify object present on the scene (i.e., object refinement). The following step is situation refinement. Once each object has been identified, this stage tries to establish a relationship between objects on the scene. The last step is threat refinement where these relationships are drawn into the future to establish possible threats. This model is depicted in Figure 7.

Under this model, all the processing is done on a central control; the sensors are only responsible for sending their data. The burden of processing is allocated only at sink, and the distributed power of sensor network is wasted. To overcome this problem we divide the model as follows. The proposed system divides the Data Fusion Process Model so that the pre-processing and object refinement is performed at the OPU and the rest of the functions are realized at the SPU. This model not only takes advantage of the distributed nature of the sensors, but also increases parallelism and speed, because tasks can be done on each sensor at the same time. The functions to be performed on the OPU are depicted in figure 8, while the SPU functions are illustrated in Figure 9.

Blind Beamforming is used in our framework for fusion and detection. The Blind Beamforming enhances the signal by processing only the array data without much information about the array [24]. Blind beamformer obtains array weight from the dominant eigenvector associated with the largest eigenvalue of the space-time sample correlation matrix. Blind beamforming with Least Mean Square algorithm uses a minimum mean square error criterion to determine the appropriate array weighting filters and is highly suitable for power aware wireless sensor networks [25-26].

This algorithm is employed in our approximation since it is power aware, it is easily adaptable to the propose data fusion model, and it does not require that all the signals are present at the same time. The pre-processing performs a fast Fourier transform (FFT) of the L sample in each sensor. FFT is performed in each sensor and only one is in charge of combining this signal. The LMS Beamforming is included in the data fusion model but is only used for the cluster head. The rest of the OPU merely performs FFT. Figure 10 illustrated the use of blind beamforming at OPU.

4. Design, Operation and Simulation of the System

The area of interest is divided into sub-areas according to a range of view of cameras. Each sub-area contains a camera and possibly other minor sensors as acoustic sensors. The cameras perform object detection at all times. Each camera has a Region Agent (RA), which manages Object Agent (OA) creation as will be described in the next section. Once an object has been detected in his area, the RA creates an OA. OA is responsible for tracking the object and sending the tracking information to the RA. Both agents reside on an OPU. OPU sends RA information to an SPU along with the image of the detected object. The SPU takes the object and tracking information and performs object classification and decomposition. For highly restricted area, we assume a node-to-cluster ratio of 4 OPUs to 1 SPU, while for low security areas we consider a ratio of 20 OPUs to 1 SPU. The ratio is established depending on the processing time. In order to preserve real time operation, the SPU only can analyze a number of OPUs that has to be determined in terms of processing time. The report of activities includes tracking information and object classification. For example an object is identified and tracked, and its information is sent to a SPU which classifies the object as a person, then performs decomposition and identifies that in the hand of the human an unknown object is present which poses a threat. The system will initiate an alarm if any of the following situations occur

- A threat is identified
- OPUs are unable to keep contact with the SPU
- OPUs are unable to communicate between them

The latest could not be a real threat; however a defective communication could result in missing a real security problem.

4.1 Cooperative Agent for Detection and Tracking

The agent framework is well suited for application to our scene understanding because it has multiple desirable characteristics [27], such as:

- It provides mechanism for binding together a set of tasks related to a particular input
- It allows a clear specification of the interface between these sets
- It facilitates an event driven process control.

Tracking people passing through an area of interest cannot be done by a single agent since the information on them is temporally and spatially distributed. Cooperating agents that collect spatial and temporal information through the entire area solves the problem.

Our approach to scene understanding incorporates agent under the following scheme. The area of interest is divided into several sub-areas in agreement with camera range view as illustrated in Figure 11. Each region corresponds to a sub-area where the camera has the best view. Each sub-area has assigned a camera and a RA. Video source is a fixed video camera with wide range of view. Object detection is done by means of background subtraction. Instead of using the RGB values of the image, a conversion to luminance values is performed. One frame is used as a reference. A 9-element vector replaces each luminance value of each pixel. The values of the new vector are the region of support of the central pixel. When the vectors have been formed, a linear dependence proof is performed. This process is repeated through the entire frame. As a result, a foreground mask containing only the moving object is obtained. This foreground mask is sent to the RA.

The RA receives the foreground mask after background subtraction process has been completed, along with the image. Then the image is segmented using the foreground mask. Each segment is sent to the OAs that have been already spawned by the RA. If any OA does not recognize a segment then a new agent is spawn to track that object. OA is responsible for updating the object model consisting of velocity, acceleration and heading based on information subtracted from several frames. When an object approaches the border of the area monitored by the RA, this agent must communicate with the proper agent to send all the information on the object to it. The RA negotiates proper handoff of moving objects leaving its area with its neighbor.

Each OA has its Tracking Database to store all the values of the model parameters. The task of the OA is to identify its assigned object, updated image segment and trajectory on it tracking database, while the RA is responsible for creation of OAs and assignment of detected object to the OA.

4.2 Region Agent Model

The RA is responsible for monitoring its area as well as coordinating OAs already assigned to the detected objects. In order to perform its activities, a RA consists of four modules: Communication Module, Object Agent Status Module, Object Agent Creation Module, and Decision Module. Figure 12 describes the agent model. Each of these modules is explained in this section.

When a new event has been detected, RA must create a new OA responsible for track and update the object model. The algorithmic description of this agent is illustrated in Figure 13.

Object Agent Status Module

The RA functions as a coordinator for all the OAs that have been created by it. In order to perform its activities, RA must know the status of the OA. When a new frame has arrived, RA is responsible for segmenting the frame. Each segment of the frame contains a detected object. The RA marks all the OAs as NO_ID to indicate that none of the OA has been able to identify the object present at the scene. When an OA recognizes its object, it sends an acknowledgment message to the RA then updates its status as ID_ACK.

Decision Module

Decision module is in charge of generating all messages to the other agents. When a new frame arrives, the OA status must be updated and messages are sent to OAs announcing the arriving of a new frame. Then, this module decides the order of transmission to the OAs. When a segment has not been identified for the OAs already created, a new OA must be spawned. When an object is approaching the border of the area, communication with the proper RA is engaged. The decision is based on the heading of the object.

Communication Module

Communication module allows the RA to exchange information with the other agents via a predefined set of messages. The type of messages and the content of the message are chosen by the decision module. The agents utilize a protocol based on the Knowledge Query Manipulation Language (KQML). KQML [28] is based on *Speech Act Theory* [29] and is a popular protocol that is being used widely for communication among agents [30-31]. All the necessary information for the correct interpretation of the message is included in the message itself. The messages used on this protocol are described as follows:

- NFA: *New Frame Arrive* message is sent to all OAs that depend on this particular RA. The message also contains a segment of the frame to be identified by the agent. OAs are expected to answer with an ACK or NAC message. ACK is an acknowledgment message indicating that a positive match has been established. While a NAC message indicate a negative match.
- ANL: *Agent Next in the List* indicates which OA is next in line. The purpose of this message is to indicate the OA to which agent must be sent the segment of the frame in case a negative match have been established
- NMS: *No More Segments* message signifies that there are no more segments to be analyzed by the OAs. This message is sent to all OAs that have been unable to identify their assigned object in the current frame.
- OAA: *Object Approaching Area* message is sent to another RA to indicate that a tracking object is drawing up in its area. This message is the start point for a negotiation of proper OA trade.

- OAT: *OA Trade* message indicates when two RAs have completed an OA trade. Trade of agents is performed as soon as a detected object is leaving the area of one of the RA and entering the area monitors by the next RA.
- NRA: *New RA* parent message is sent to an OA trade to another RA when the object that has been tracked as leaving the area. The purpose of the message is to indicate the OA that it has a new coordinator.
- PING: This message is used when no communication can be established with another agent. A ping message is sent to the agent to find out if the agent is still active. Agent responds to this message with an ACK message. No further communication is done with that specific agent until an ACK message has been received
- ACK: *Acknowledgement* message is a response to a PING message sent for another agent.

4.3 Object Agent Model

An OA is responsible for determining if its assigned objects appear on the scene. Also, it is responsible for updating the model and let the RA know that a positive match was established. To execute its task, the OA is composed by communication, object matching, update and decision module. This agent also contains a tracking database to store all the previous values of the velocity, acceleration and heading of the detected object. The OA model is depicted in Figure 12. The Update module is responsible for updating the new object parameters in the tracking database.

The Decision Module generates the message to communicate with the RA. The OA must inform when a positive match can be established. Also, the Decision Module chooses when the update process must be performed. Algorithm of OA is depicted in Figure 14.

Object Matching Module

The Object Matching Module recognizes if the segment contains the assigned object. The decision is taken using the Mahalanobis distance. Mahalanobis distance is a technique to determine similarity between a set of values and an unknown sample [32]. The Mahalanobis distance takes the information of the variance and covariance between variables. That means that the interaction between variables and the range of acceptability is used to determine the similarity between the two sets of values. The distance is calculated by the equation 1. Where I represent the matrix containing the values of the image of the segment, m_t is the vector of the means of the variable of the detected object and S_t is the with-in group covariance of the detected object.

$$D_M(I) = (I - m_t)S_t^{-1}(I - m_t) \qquad (1)$$

Communication Module

OA must communicate with the RA and the OA. The predefined messages that are sent are described as follows:

- NSG: *New Segment* has been sent to the next OA for matching process. If the agent does not recognize the segment, then it must send the segment to the next agent. RA is responsible for indicating which agent is next in line.
- EXP: *Expiration message* is sent to the RA to inform that the OA has decided to terminate. The decision comes if after three frames the object cannot be identified.
- ACK: *Acknowledgement* message is a response to a PING message sent for another agent. Its also use to indicate that the object has been positively match
- NAC: *No Acknowledgment* message indicates that a negative match of the assigned object has been established. When the RA receives this message, it sends the next segment until no segments are left.

Ontology for the Multi-Agent System

The simulation of the cooperative agents is done by ZEUS toolkit [33]. ZEUS toolkit provides tools for simulation and development of cooperative agents in form of Java classes' package. ZEUS provides different default role modeling to implement the functionality inherent to a multi-agent application. The first stage on ZEUS implementation is decided which role model suits the given implementation. The role model for this particular application is Shared Information Space. This model allows each agent to be a publisher and subscriber simultaneously.

The responsibilities for every publisher include sending information to subscribers, responding to information sent for subscribers and performing its specific activities. Subscribers must respond to the publisher and perform its activities. RA publishes image segments and each OA takes an image segment and identifies if corresponds to its assigned object. If a part of the object cannot be identified by any OA, then the RA creates a new OA for that particular object. The ontology is the shared understanding of the interest domain. Agents communicate between each other to cooperatively solve a designated problem. The communication is performed via messages. Each message contains parameters that posses a meaning in the problem domain. In order for the agents to understand these messages, they must share a common knowledge. The definition of the ontology is defined to be a set of facts. Table 2 present the set of fact defined for this application.

4.4 Cooperation Algorithm

Agent must cooperate to be able to detect and track all moving objects on the scene. Figure 15 describes the state transition of the cooperation state between the RA and the OAs. An RA announces that a new frame has arrived to all OAs depending on the arrival (state 1). After segmentation of frame on the detected objects, the RA sends a segment to the first agent on the list (state 2).

This agent must try to recognize if the segment contains its assigned object. If the agent recognizes the object (state 3), update procedure is in order (state 5). However if the agent is incapable of distinguishing its object on the segments, then hand it off to the next agent (state 4). OA knows who is next in line because the RA provides that information via messages. If the last agent is reached and the RA still has some segments a new OA is spawned (state 7). New agent does not perform object matching, but automatically goes to update the information (state 5). When all the segments have been analyzed by the OAs, some of them may not have successfully matched the assigned object on the scene; therefore RA must inform them that no more segments are available. RA locates the unsuccessful agents and sends them a message indicating that all segments have been analyzed (state 9). The OA updates the number of unsuccessful attempts (state 10) to decide when to terminate its tracking activity.

Cooperation between RAs is also necessary for proper hand off when an event is leaving the area of one RA and entering the area of the next agent. The RA knows its area and the RAs that manage the neighboring area at each of the borders. This information is necessary for the decision module; so that the agent can determine which area the event is entering. When a RA R_j determines that an object is approaching its border area, it sends a message to all RAs with heading information, which is the state 2 illustrated in Figure 16. RAs use heading information to decide if the event is entering its area. If the decision is negative, then the agent sends a negative response to agent R_j. On the other hand, if the event is actually heading towards the area controlled by agent R_i, then hand off (for the proper OA) is performed between agents R_i and R_j state (5). That ends the negotiation between agents for proper exchange of crossing area events.

In order to clarify how the information flows in the system, Figure 17 shows a sample scenario. There are already 3 OAs tracking different events. The foreground mask arrives to the RA then the agent produces four segments. The first segment is sent to OA 1. The agent does not recognize the object, therefore sent the segment to the next object in line, which is agent 2. The RA sends the next segment to agent 1. This time agent 1 recognizes the event and sends an acknowledgment message to RA. Agent 2 sends segment 1 to agent 3, while RA send it segment 3. Agent 2 recognizes the event and sends acknowledgment message to RA. Next, agent 3 recognize segment 1 and sends the acknowledgment to RA. Then all existing agent have recognized its event, but a segment still available. Then RA creates a new OA. This cooperation scheme allows matching process to be pipelined, reducing time of processing.

4.5 Object Detection, Classification and Threat Detection

A comparison between background subtraction methods [34-35] shows that *Mixture of Gaussian* (MG) and *Wronskian Change Detector Model* (WCD) are the most promising methods. However, MG is computationally intensive, therefore not suitable for our approach. WCD requires converting each pixel of an image into 9-dimensional vectors. The vector replaces the center pixel of a region of support. The component vector corresponds to the luminance values stored in each pixel of the image. A change in the region means that the luminance vectors are linearly independent. A simple and rigorous test for determining the linear dependence of vector is the Wronskian determinant. The Wronskian is easily calculated and results in:

$$W\left(\frac{x_i}{y_i}\right)=\frac{1}{n}\left(\sum_{i=1}^{n}\frac{x_i^2}{y_i^2}-\sum_{i=1}^{n}\frac{x_i}{y_i}\right)=0 \tag{2}$$

Our simulation accepts frames in JPEG format from the camera where each frame size is 640 ×480 pixels images. Each pixel is an 8-bit value ranging from 255 to 0. The original image is in RGB format that is converted to a luminance values. The resulting foreground mask has zero values for elements on the background and 255 values for detect moving objects.

The camera generates 30 frames per second. However it is not really useful to subtract the background for each of these frames. Instead, our simulation took the first and the last frame for each second. This interval reduces timing and power consumption because the operation is performed only once each second rather than 30 times per second. Outdoor image tracking of vehicles requires a smaller interval of time compared to detecting humans.

Background Subtraction algorithm requires vector generation of 9 elements, however the vector dimension could vary from 3 × 3 region to a 5 × 5 or even greater values. Our simulation considers only regions of 3 × 3 and 5 × 5. However, the detection improvement using a 5 × 5 windows are not found to be extra useful. Our simulation reports that a greater region almost double the processing time, therefore a fixed region of 3 × 3 is considered enough for our simulation. Figure 18 shows result for an outdoors image using window size of 3 × 3 and 5 × 5. Figure 19 illustrates simulation results for background subtraction of an indoor image. A fixed camera embedded in a sensor is a source of video. The results for two different size windows are the same in this case.

Background subtraction algorithm must be robust against change of global illumination. In order to measure the performance of the Wronskian detector, images with different luminance average values were tested. The results are shown in Figure 20 for indoor and outdoor scenes. Luminance average values give a measure of the global illumination of the scene. A factor of 0.47 was obtained at noon when the brightest light can be observed for outdoor. Brightest indoor scene has a factor of 0.31. Since the system will be deployed on building and parking lots, a medium change of illumination can be assumed. Based on the results, Wronskian detector is sufficiently robust for our application. However, further improvement could be achieved by including an illumination compensation block before background subtraction.

Object classification can be done by two schemes: neural networks or statistical techniques. Statistical techniques offer a low complexity approach, independence of orientation and resilience to noise. However ANN had proved better accuracy in classification than statistical techniques [37]. In our set up, once the object has been identified, it is passed through the next stage, which is object decomposition. Our approach uses linear Support Vector Machine (SVM) to decompose the object for threat identification purposes.

Ongoing Work

Threat recognitions must be done by analyzing behavior of detected objects. There are two main sources of attack, humans and vehicle. A human displays criminal activity such as hitting or gun pointing, while a car poses a threat when its velocity exceeds the limits set for the area. Therefore, a spatio-temporal pattern detection approach must be employed. Statistical learning techniques such as Hidden Markov Model can be employed to detect criminal pattern activity. However, the nature of these events imposes several challenges such as small and imbalance training sets. In view of the fact that criminal activity is a rare event, the amount of training data is small, furthermore is desirable that new criminal pattern can be detected without explicit retraining. Since the amount of data containing a threat is smaller than the data containing normal activity, the training set may be uneven.

Statistical learning models can be classified as generative or discriminative. Generative techniques such as Hidden Markov Model learn a class density for each class. When a new event is present these techniques select the most likely class by using a Maximum a posterior class. On the other hand, discriminative models estimate the posterior for each class without modeling the class density. Solutions to these problems have been proposed in [18, 39].

Coupled Hidden Markov and Hidden Markov models have been applied towards behavioral analysis, agent have been employed to provide training set for this models [18]. This effort has no consider vehicle models that analyze the behavior of such object. While a Support Vector Machine has been proposed as a solution to uneven and small training sets [39].

The information that our threat detection unit has is the tracking information gathered for our Multi-agent system that is further classified as a human, vehicle or animal as shown in Figure 21. Our efforts are on developing models for humans and vehicle that include actions over an extended period of time. We are analyzing several discriminative models to perform the threat analysis.

5. Conclusion

A distributed scene understanding architecture has been presented. The architecture consists of several OPUs distributed on the area under surveillance. Each OPU is attached to a camera. OPUs are responsible for detecting and tracking moving objects on the scene. Object detection is performed by background subtraction, while cooperative agents track detected objects. This information is sent to a SPU, which will classify and decomposed objects. Then, the SPU determines if the scene contain a threat pattern and perform appropriate action. This architecture represents an effort to automate surveillance systems. Tasks are distributed in the network, so the SPU are not loaded of scenes that may not contain useful information. The specific focus of this paper is in presenting the Mutli-agent system level architecture.

Acknowledgements

The authors acknowledge the support of the U.S. Department of Energy (DoE), EETAPP program DE97ER12220, the Governor's Information Technology Initiative, the DoE award DE-FGO2-04ER46136 and the Louisiana Board of Regents contract DOE/LEQSF (2004-07)-ULL, and the support of National Science Foundation NSF, INF 6-001-006.

References

[1] A. R. Dick & M.J. Brooks, 2003. Issues in Automated Visual Surveillance. In *Proceeding of VIIth Digital Image Computing: Technique and Applications*, Sydney10-12 Dec. 2003, 195-204

[2] S. S. Iyengar & R. R. Brooks, 2005. An Overview, in *Distributed Sensor Networks*, Chapman & Hall/CRC, 3-10

[3] M.A. M. Vieira, C. N. Coelho, D.C. da Silva, J.M. da Mata, 2003. Survey on Wireless Sensor Network Devices. *In Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, Lisbon, Portugal, 16-19 September 2003, IEEE, Vol. 1, 537-544

[4] S. Meninger, J.O. Mur-Miranda, R. Amirtharajah, A.P. Chandrakasan & J. Lang, 2001. Vibration-to-Electric Energy Conversion. *IEEE Transactions on VLSI Systems*, 9(1), 64-76.

[5] A. Lal, J. Blanchard, 2004. The Daintiest Dynamos. *IEEE Spectrum*. 41(9), 36-41

[6] D. D. Wentzloff, B. H. Calhoun, R. Min, A. Wang, N. Ickes & A. P. Chandrakasan, 2004. Design considerations for next generation wireless power-aware microsensor nodes. In *Proceedings of 17th International Conference on VLSI Design*. Mumbai, India. January 5-9 2004. Vol. 1, 361-367

[7] A. Sinha & A. Chandrakasan, 2001. Dynamic Power Management in Wireless Sensor Networks, *IEEE Design & Test of Computer*. 18(2), 62-74

[8] N. Jennings & M. Wooldridge, 1998. Applications of Intelligent Agents. *Agent Technology: Foundations, Applications, and Markets*, 3-28.

[9] H.S. Nwana & D. T. Ndumu 1999. A Perspective on Software Agents Research. *The Knowledge Engineering Review*. 14(2), 1-18.

[10] R. Nair, M. Tambe, S. Marsella & T. Raines, 2004. Automated Assistant for Analyzing Team Behaviors *Autonomous Agents and Multi-agents Systems Journal*. Kluwer Academic Publishers, 8, 69-11

[11] N. Pissinou, K. Makki, M. Hong, L. Ji & A. Kumar, 1997. An Agent Based Mobile System. In Proceeding of *16th International Conference on Conceptual Modeling*, Los Angeles, USA November 3-5 1997, 361-374.

[12] S. Aknine, S. Pinson & M. F. Shakun, 2004. An Extended Multi-Agent Negotiation Protocol. *Autonomous Agents and Multi-agents Systems Journal*. Kluwer Academic Publishers Vol 8, 5-45

[13] M. Palaniappan, N. Yankelovich, G. Fitzmaurice, A. Loomis, B. Haan, J. Coombs & N. Meyrowitz, 1992. The Envoy Framework: An Open Architecture for Agents. *ACM Transaction on Information Systems*. ACM 10(3), 233-264.

[14] L. Hodge & M. Kamel, 2003. An Agent-Based Approach to Multisensor Coordination. *IEEE Transactions on Systems, Man, And Cybernetics—Part A: Systems and Humans*. IEEE, 33(5), 648-662

[15] R. T. Collins et al. 1999. A System for Video Surveillance Monitoring. *Carnegie Mellon University*. Report No. CMU-R1-TR-00-12.

[16] E. Shakshuki & Y. Wang 2003. Using Agent-Based Approach to Tracking Moving Objects. In *Proceedings of 17th International Conference on Advance Information Networking and Application,* Xi'an, China, March 27 – 29 2003. IEEE, 578-581

[17] P. Remagnino, A.I.Shihab & G. A. Jones 2004. Distributed Intelligence for a Multi-Agent Visual Surveillance, *The Journal of the Pattern Recognition Society* Elsevier, 37(4), 675-689.

[18] N.M. Oliver, B. Rosario & A.P. Pentland, 2000. A Bayesian Computer Vision System for Modeling Human Interactions, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22 (8), 831-843.

[19] L. Rabiner & B. Juang, 1986. An Introduction to Hidden Markov Models, *IEEE ASSP Magazine*, 3(1), 4 – 16.

[20] L.R. Rabiner, 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, 77(2), 257 – 286.

[21] M. Brand, N. Oliver, A. Pentland, 1997. Coupled Hidden Markov Models For Complex Action Recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan 1997, 994-999.

[22] H. Buxton, 1997. Advanced Visual Surveillance Using Bayesian Networks. *IEE Colloquium on Image Processing for Security Applications*, Digest No. 1997/074, March 10 1997, 9/1-9/5.

[23] J. Llinas & D. Hall, 1998. An Introduction to Multi-sensor Data Fusion. In *Proceeding of IEEE International Symposium on Circuits and Systems,* Monterey, USA. 31 May – 3 June 1998, vol. 6, 537-540.

[24] J.C. Chen, K. Yao, & R.E. Hudson, 2002. Source Localization and Beamforming. *IEEE Signal Processing Magazine*, 19(2), 30 – 39.

[25] W. Heinzelman, A. Sinha, A. Wang & A.P. Chandrakasan, 2000. Energy-Scalable Algorithms and Protocols for Wireless Microsensor Networks. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Geneva Switzerland, 5-9 June 2000. IEEE, vol.6, 3722 - 3725.

[26] A. Wang & A. P. Chandrakasan, 2002. Energy-efficient DSPs for Wireless Sensor Networks, *IEEE Signal Processing Magazine*, 19(4), 68-78.

[27] M. Dietl, J-S Gutmann, & B. Nebel, 2001. Cooperative Sensing in Dynamic Environments. In *Proceeding of International Conference on Intelligent Robots and Systems*, Hawaii, USA, 29 Oct.-3 Nov 2001. IEEE/RSJ, vol.3, 1706 - 1713.

[28] T. Finin, D. McKay & R. Fritzon, 1992. An Overview on KQML: A Knowledge Query and Manipulation Language. *University of Maryland Computer Science Department*, Technical Report No.CS-94-12.

[29] J. R. Searle, 1970. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

[30] M.N. Huhns & M.P. Singh, 1998. *Readings in Agents*. San Francisco, Morgan Kaufmann.

[31] G. Weiss, 1999. *Multiagent System: a Modern Approach to Distributed Artificial Intelligence*. Cambridge, Massachusetts, MIT Press.

[32] T. Kamei, 2002. Face retrieval by an adaptive Mahalanobis distance using a confidence factor. In *Proceedings of International Conference on Image Processing*, 2002. Vol. 1, I-153 - I-156.

[33] J.C. Collis, D.T. Ndumu, H.S. Nwana & L.C. Lee, 1998. The ZEUS agent building toolkit. BT Technology Journal, 16(3), 60-68

[34] S-C. S. Cheung & C. Kamath 2004. Robust Techniques for Background Subtraction in Urban Traffic Video. In *Proceedings on Video Communication and Image Processing*, San Jose California, USA January 2004, SPIE Electronic Imaging. UCRL-JC-153846-ABS, UCRL-CONF-200706.

[35] E. Durucan & T. Ebrahimi, 2001. Change Detection and Background Extraction by Linear Algebra. *Proceedings of the IEEE*, 89(10), 1368-1381.

[36] T.Matsuyama & N. Ukita 2002. Real-time Multitarget Tracking by a Cooperative Distributed Vision System. *Proceedings of the IEEE*, 90(7), 1136- 1150.

[37] B.D. Maciel & R.A. Peters, 2000. A Comparison of Neural and Statistical Techniques in Object Recognition. In *Proceeding of IEEE International Conference on Systems, Man and Cybernetics*, October 8-11 2000. IEEE, Vol. 4, 2833 - 2838.

[38] B. Heisele, 2003. Visual Object Recognition with Supervised Learning. *IEEE Intelligent Systems*. 18(3), 38-42.

[39] Z. Dimitrijevic, G.Wu & E.Y. Chang, 2003. SFINX: A Multi-sensor Fusion and Mining System, In *Proceedings of the IEEE Pacific-rim Conference on Multimedia*. Singapore, 15-18 December 2003, Vol. 2, 1128-1132

Ruth Aguilar-Ponce obtained a Bachelors degree in Computer Science from the Autonomous University of Puebla, Mexico in 1997. She obtained the M.S. degree in Electronics from the National Institute for Astrophysics, Optics and Electronics, Mexico in 1999. She joined the Center for Advanced Computer Studies at the University of Louisiana at Lafayette, where she obtained the M.S. in Computer Engineering in 2004. Currently she is working towards her Ph.D. degree. Her research interest includes sensor networks, software systems, communication protocols, system on a chip, signal and image processing.

Ashok Kumar obtained a B.Tech. degree in Computer Science & Engineering from the Institute of Technology, Varanasi, India in 1990. From 1990 to 1994, he worked with Uptron India Limited, Lucknow in the capacities of Engineer and Senior Engineer. He obtained M.S. and Ph.D. degrees from the Center for Advanced Computer Studies (CACS) at the University of Louisiana at Lafayette in 1997 and 1999, respectively. After getting his Ph.D., Dr. Kumar worked with Cadence Design Systems, USA for nearly four years in the capacity of a Senior Member of Technical Staff for R&D operations. Currently he is a Research Faculty at CACS. Dr. Kumar has served on the program committees of five conferences as well as on the proposal-evaluation committees of the National Science Foundation.

J.Luis Tecpanecatl-Xihuitl received the B.E degree in Electrical Engineering from the Autonomous University of Puebla, Mexico in 1997 and the M.S. degree in Electronics from the National Institute for Astrophysics, Optics and Electronics, Mexico in 2000. He is currently a Ph.D. candidate in Computer Engineering at the Center for Advanced Computer Studies, University of Louisiana at Lafayette. His research interests focus on multi-standard digital receivers, VLSI digital processing systems and image processing

Magdy Bayoumi is the Director of the Center for Advanced Computer Studies and Department Head of Computer Science Department, University of Louisiana at Lafayette. He is also the Edmiston Professor of Computer Engineering and Lamson Professor of Computer Science. Dr. Bayoumi received the B.Sc. and M.Sc. degrees in Electrical Engineering from Cairo University, Egypt; a M.Sc. degree in Computer Engineering from Washington University, St. Louis; and a Ph.D. degree in Electrical Engineering from the University of Windsor, Canada. Dr. Bayoumi has served as the Chair for several IEEE societies and conferences, and he is a Fellow of the IEEE. Dr. Bayoumi won the UL Lafayette 1988 Researcher of the Year award and the 1993 Distinguished Professor award at UL Lafayette.

List of Tables and Figures (each supplied on a separate page)

List of Tables

List of Figures

Table 1
Summary of Previous Works and Comparison

	Deployment/ Distribution of Units	Method of Detection	Ability to Track Multiple Objects	Automated Tracking	Agent Based System	Connection Style	Susceptibility to Network Attack
[15]	Cameras are associated to a computer	Implemented by adaptive background subtraction and three-frame differencing. (Require storage of three frames, therefore a large amount of memory)	Restricted	No	No	Wired /Wireless	Susceptible to attack. OCU functions as an arbiter, if OCU is destroyed, the network will fail
[18]	Fixed cameras without processing unit.	Eigenbackground subtraction. (Requires large amount of memory and computational power.)	Yes	Yes	No	Wired	Susceptible to attack. Cameras send information to a central processing unit.
[16]	A moving object is equipped with a GPS receiver	Realize by GPS detection. (GPS information is not always available / not carried by object of interest)	Restricted	Yes	Yes	Wireless	Very prone to attack. GPS information is not available for all objects.
[17]	Fixed cameras without processing unit.	Performed by classification of moving and non-moving pixels. (Weak against illumination changes)	Yes	Yes	Yes	Wired	Susceptible to attack. Cameras send information to a central processing unit.
Ours	Each OPU contains a camera and a processing unit. Clusters are deployed.	Background subtraction implemented by Wronskian Detector (Robust against illumination changes and computationally efficient)	Yes	Yes	Yes	Wireless	Least Susceptible. The approach is distributed therefore it is less susceptible to attacks

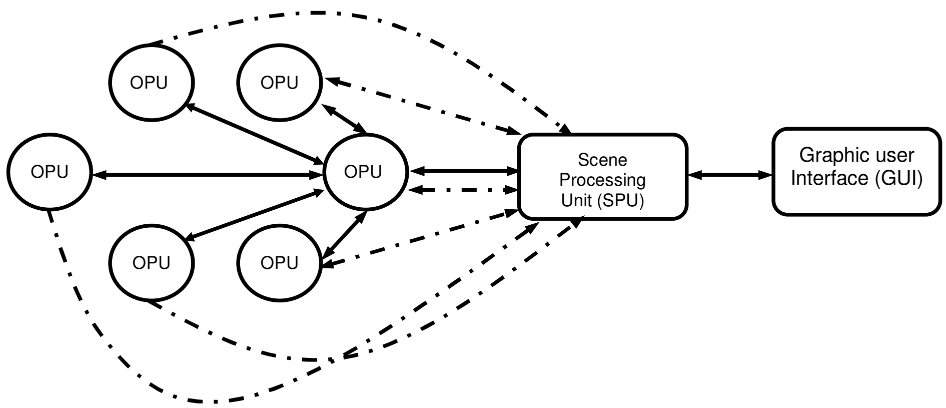


Fig. 1. Overall Proposed Architecture

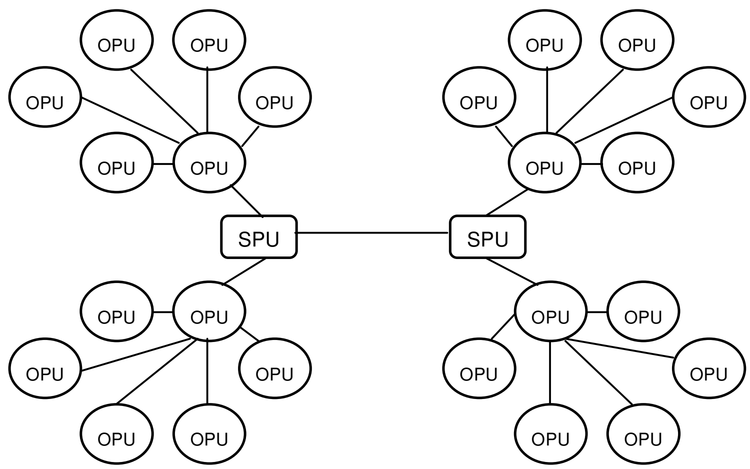


Fig. 2. Connectivity of the Whole System (All Connections are Wireless)

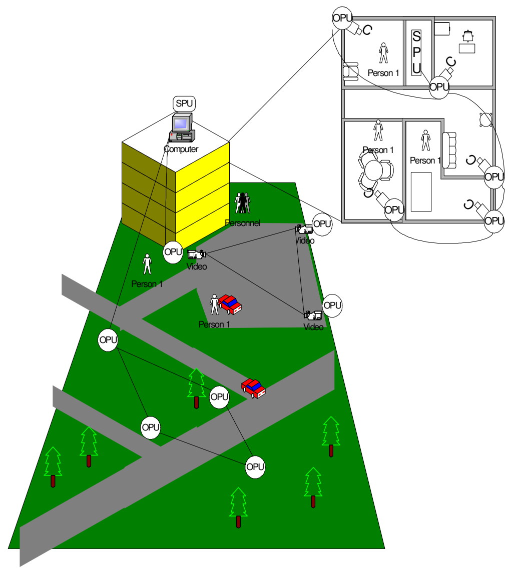


Fig. 3. Indoor and Outdoor Deployment of OPUs and SPU

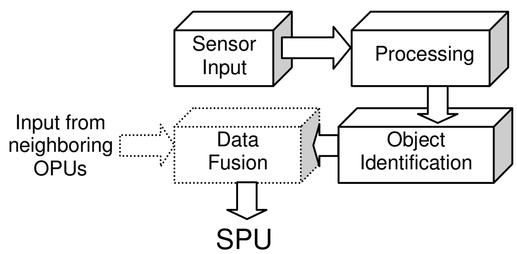


Fig. 4. Basic Functions Performed at General OPU

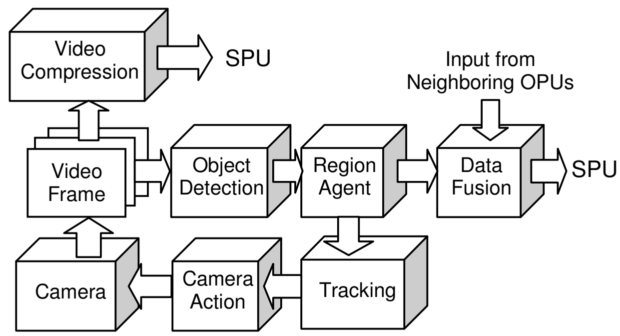


Fig. 5. General Functions Performed at OPU

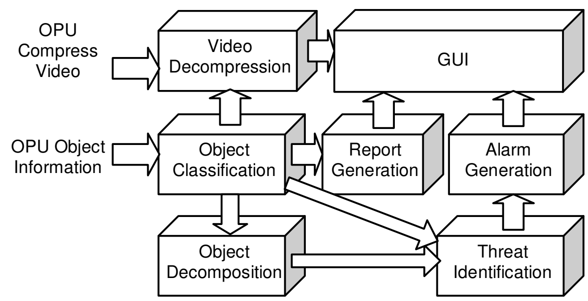


Fig. 6. Functional Diagram of SPU Tasks

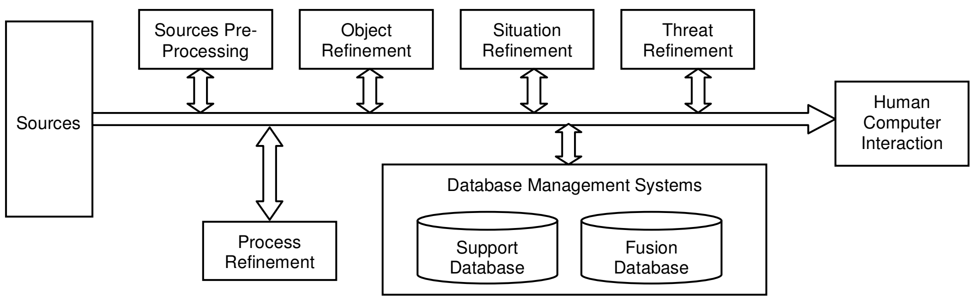


Fig. 7. Top Level Data Fusion Process Model

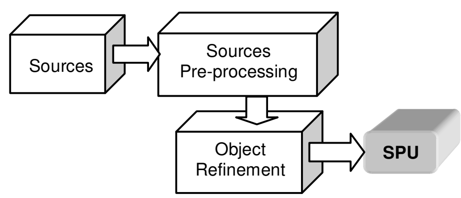


Fig. 8. Data Fusion at the OPU

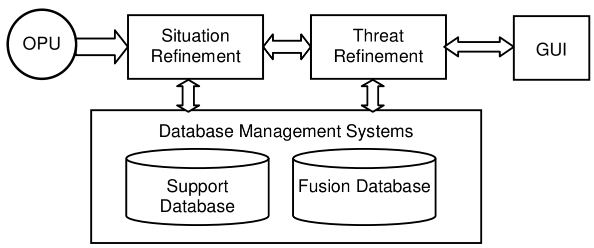


Fig. 9. SPU Data Fusion Flow

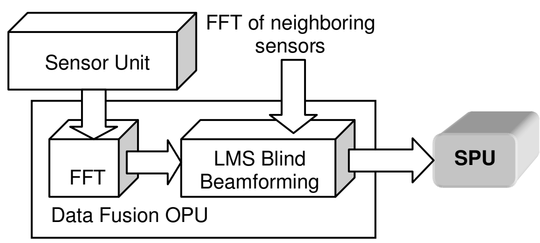


Fig. 10. Data Fusion Using Blind Beamforming on OPU

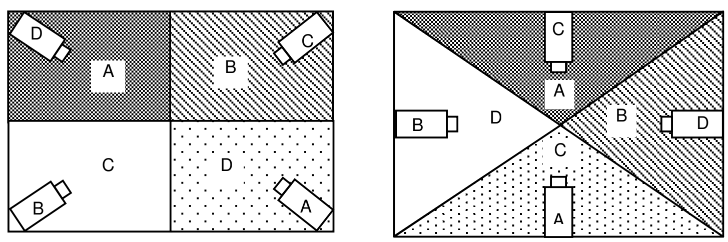
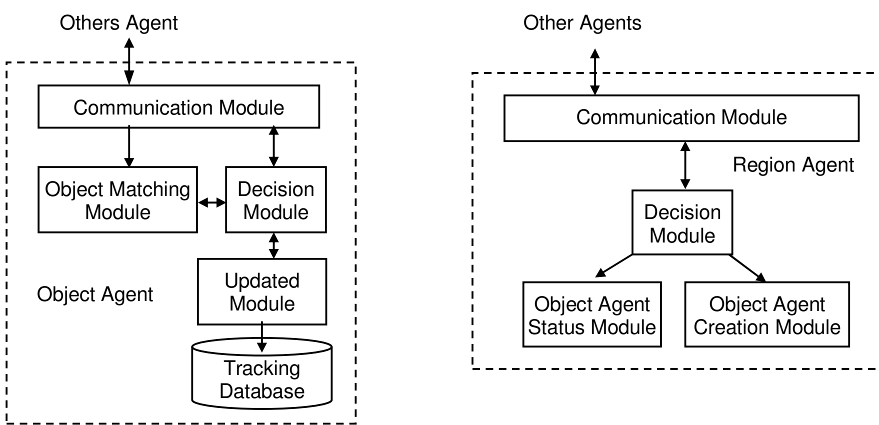


Fig. 11. Segmentation of the Area of Interest for Two Different Camera Dispositions



a. Object Agent b. Region Agent

Fig. 12. Agents Models.

```
Region Agent
BEGIN
/* Let A be the set of OA of RA Rj */
Initially A ← {} /* Empty set */

Segmentation of f_k into segments s_i
Reset of the status on the list of agents A.

FOR each a_j in set A
   FOR each segment s_i of frame f_k
        SEND segment s_i to OA a_j
        WAIT (Object Agent a_j answer)
        IF ACK is received
           THEN Update Agent status and border distance,
                Marked segment as identified
           ELSE Send a next agent in list
   END FOR
END FOR

/* Spawn new Object Agents */
IF segments cannot be identified THEN
   BEGIN
        FOR each unidentified segment u_s
            SPAWN (agent a_j+1) assigned unidentified segments
            Update border distance
        END
   END
/* Notify each unsuccessful agent that no more segments
   are available for analysis */
FOR each agent a_j that cannot identify its object
    SEND notification of no more segments to a_j
END

/* Object Agent trade */
FOR each a_j in set A
   IF border distance < threshold THEN
        SEND(object approaching area message, all RA R_j)
        WAIT(for answer for all Region Agents R_j)
        IF positive answer THEN
           Trade Object Agent a_j
   ENDIF
END FOR
```

Fig. 13. Region Agent Algorithmic Description

```
Object Agent
BEGIN

WAIT (message from Parent)

IF message is segment s_i THEN
 BEGIN
   MD = Mahalanobis_distance(s_i, assigned object)

   IF segment is identify THEN
    BEGIN
     SEND(acknowledgment message to RA)
     UPDATE(object_model)
    END
  ELSE
    BEGIN
     SEND(no acknowledgment to RA)
     WAIT(Next in list agent)
     SEND(segment s_i to next agent)
    END

  IF distance_border < threshold THEN
   WAIT(new parent from RA)
   NEW_PARENT(RA R_j)
  END
 END

IF message is no more segments to analyze THEN
   Increment number of negative id

IF negative id > threshold_id THEN
   SEND (to parent decision on terminated)
   WAIT (Parent permission)
END
```

Fig. 14. Object Agent Algorithmic Description

Table 2
Fact Definition

Fact	Attributes	Default	Meaning
Image_Segment	Segment: Matrix containing a image segment		Segment containing the object to be match
Position_on_Frame	Coordinates x, y: integer		Position coordinates of the upper left corner of the segment
Distance_to_Border	Distance: real		Distance of the object to the border
Approaching_Border	Approach: Boolean	False	Flag for object approaching border area
Velocity	Velocity: real		Velocity of the detected object
Acceleration	Acceleration: real		Acceleration of the detected object
Heading	Heading: real		Heading of the detected object
Positive_id	ID: Boolean	False	Positive match of the segment with the assigned object
Number_NegID	NO_ID: integer	0	After all the segments have been analyzed by the agent, increments this value to indicated that its object was not present on the current scene
NoSegment	NoSeg: Boolean	False	Indicated to Object Agent that there are no more segments to analyze
ObjectAgentID	Name: string		Name of the Object Agent
RegionAgentID	Name: string		Name of the Region Agent

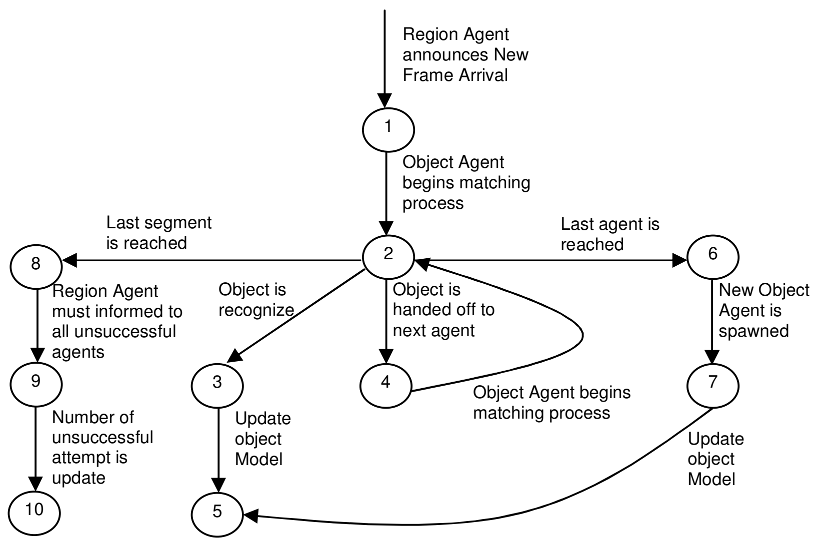


Fig. 15. State Transition Graph of the Cooperation between Region and Object Agents

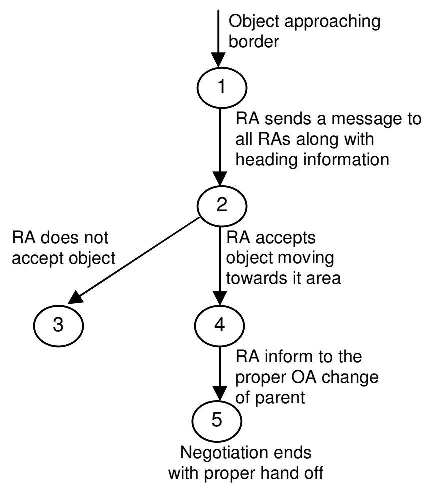


Fig. 16. State Transition Graph of the Cooperation between Region Agents

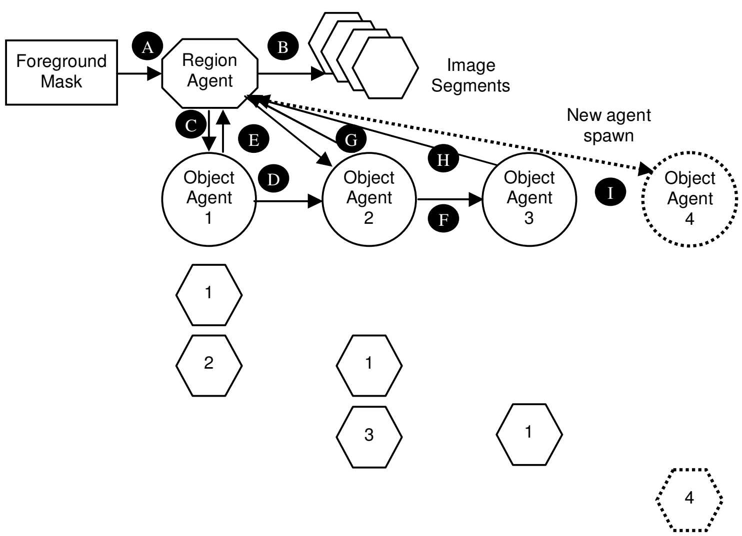
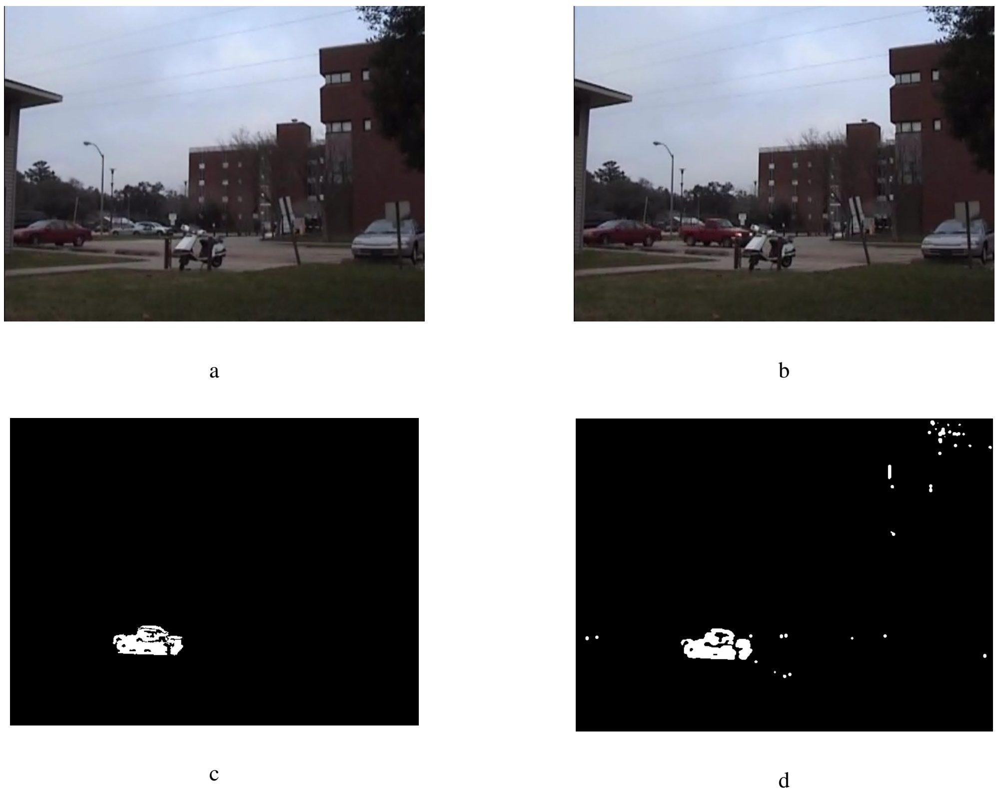


Fig. 17. Data flow of Agent System



a b

c d

Fig. 18. Simulation Results for Background Subtraction of an Outdoor Image. a) Background Image, b) Image Containing a Moving Object (Red Truck), c) Results with a 3×3 Window and d) Results with a 5×5 Window

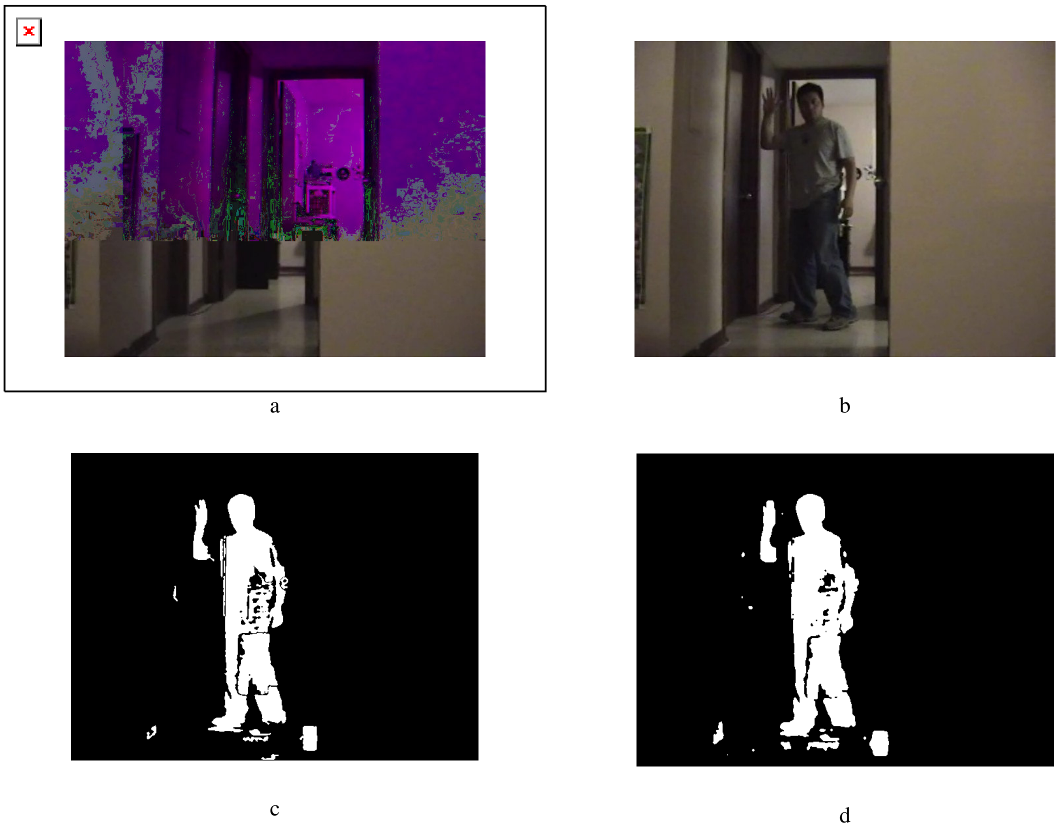
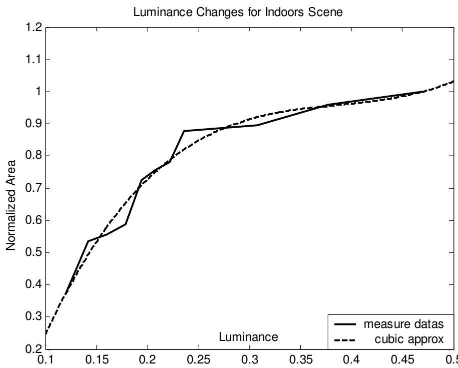
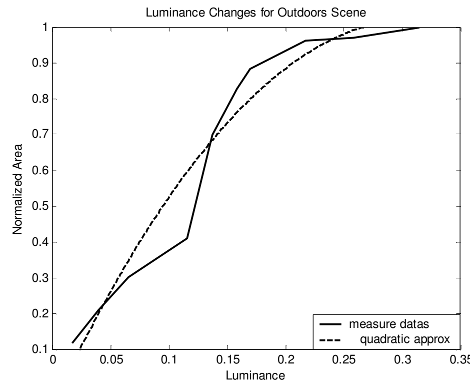


Fig. 19. Simulation Results for Background Subtraction of an Indoor Image. a) Background Image, b) Image Containing a Moving Object (Person), c) Results With A 3 × 3 Window And d) Results With A 5 × 5 Window



a



b

Fig. 20. Detection Performance for Different Luminance Values. a) Results for Indoor Scenes, b) Results for Outdoor Scenes

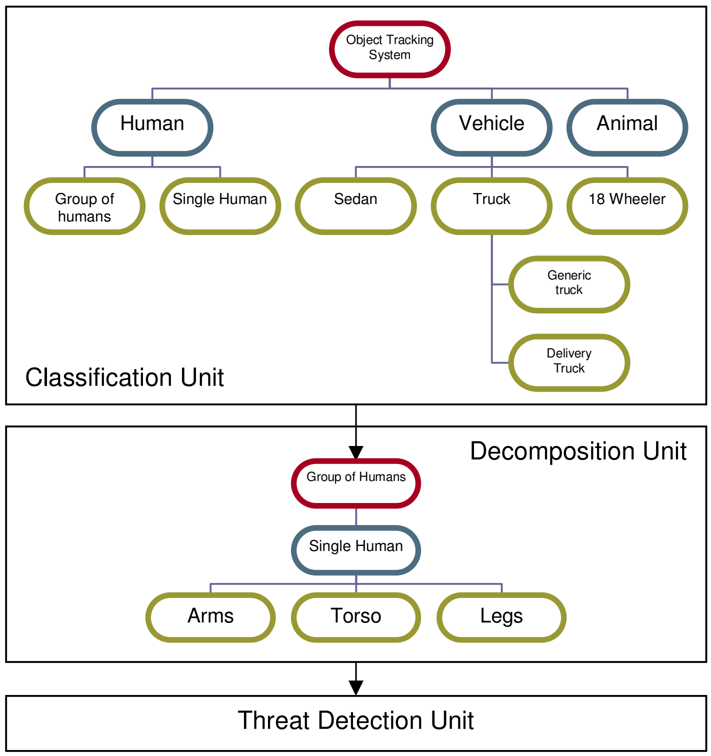


Fig. 21. Classification of Detected Objects