Chapter XI

Automated Object Detection and Tracking for Intelligent Visual Surveillance Based on Sensor Network

Ruth Aguilar-Ponce, University of Louisiana - Lafayetta, USA

Ashok Kumar, University of Louisiana - Lafayetta, USA

J. Luis Tecpanecatl-Xihuitl, University of Louisiana - Lafayetta, USA

Magdy Bayoumi, University of Louisiana - Lafayetta, USA

Mark Radle, University of Louisiana - Lafayetta, USA

Abstract

The aim of this research was to apply an agent approach to a wireless sensor network in order to construct a distributed, automated scene surveillance. A wireless sensor network using visual nodes is used as a framework for developing a scene understanding system to perform smart surveillance. Current methods of visual surveillance depend on highly trained *personnel to detect suspicious activity. However, the attention of most individuals degrades after 20 minutes of evaluating monitor-screens. Therefore, current surveillance systems are prompt to failure. An automated object detection and tracking was developed in order to build a reliable visual surveillance system. Object detection is performed by means of a background subtraction technique known as Wronskian change detection. After discovery, a multi-agent tracking system tracks and follows the movement of each detected object. The proposed system provides a tool to improve the reliability and decrease the cost related to the personnel dedicated to inspect the monitor-screens.*

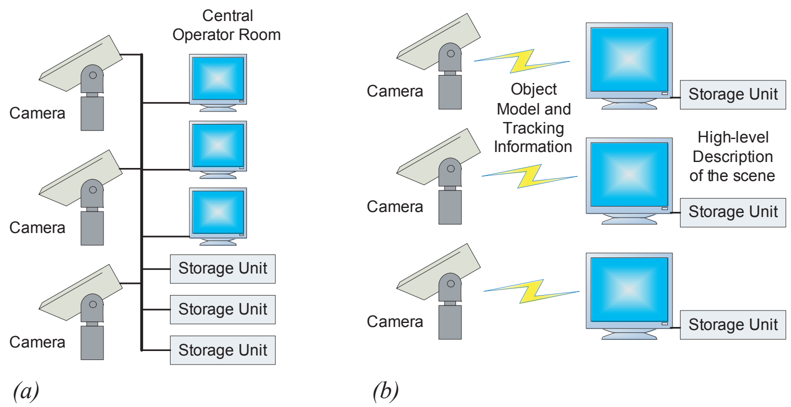
The Need for Automated Surveillance Systems

Automated visual surveillance is becoming an increasingly interesting topic for the scientific community because of the changing security needs. The need for developing computer systems that can provide enough information to take rapid action against security threats is greatly felt. Typically, visual surveillance systems 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 activities. A high-level sketch of such a system is illustrated in Figure 1a. However, with the increasing number of cameras to monitor a huge number of installations of interest, there is simply not enough pairs of eyes to keep track of all information. Moreover, a recent study concludes that after 20 minutes of evaluating monitor-screens, the attention of most individuals degrades to below acceptable levels (Green, 1999). Another factor to be considered is the cost because cameras are cheap and ubiquitous but the personnel required to analyze them are highly expensive. Therefore, the video captured by the cameras serves more as an archive to refer to after an event has occurred. A current trend in the research is to design *smart surveillance systems* capable of preventing untoward incidents rather than investigating after the incidents have occurred. However, there are many challenges to be overcome before a reliable automated surveillance system is realized (Dick, 2003). 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, reliable object detection, and trustworthy identification of individual.

There are several events that a smart surveillance system has to detect in real-time such as motion, abandoned object alert, object removal, and observation of any other abnormal activity or behavior. Motion detection alerts when objects are moving in the specified zone. Also, the system must identify the characteristics of the motion such as velocity, acceleration, and direction of movement. Another event of interest is an abandoned object that constitutes a potential threat such as bombs. It is also of interest to investigate when an object has been removed from the area under surveillance such as expensive equipment being stolen. The system must be capable of alerting against behavior that deviates from the norm such as a vehicle going over or under the speed limits in a parking lot.

Two core issues for *automated surveillance* are *object detection* and *tracking*. Surveillance cameras provide video stream that suffer from low resolution and low frame rate. Moreover,

Figure 1. (a) Typical surveillance system; (b) proposed approach to surveillance systems



the quality of the video depends on the lighting conditions. Also, suspicious activity must be detected at the time of occurrence, therefore, object detection must be performed in real-time. These characteristics make object detection a challenge. Object detection may be performed by *background subtraction* and *optical flow* techniques. The optical flow is capable of detecting object movement even when the background is also moving. However, this technique is computationally complex and resource demanding, hence, it is difficult to be implemented in real-time. On the other hand, background subtraction is more suited to detecting movements or changes on the scene, yielding to a low complexity implementation. A fixed background is required for this type of detection. The 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 the previous frames. Next, the comparison between input frame and the background model is performed. As a result, a foreground mask is built up.

In a surveillance system, several cameras could capture the same object. Therefore, cooperation among them has to be achieved in order to track such objects over an extended distance and time. Since tracking has to be performed over an extended period of time, lighting conditions and background variations arise as problems to be solved. A smart surveillance system must be capable of continuously tracking the identity, location, and activity of people and vehicles within the supervised area (Hampapur, 2003).

This chapter is organized as follows. The next section describes the previous research efforts on automated visual surveillance. Then, we explain the basis of our approach and discuss our proposed solution to object detection and tracking. We talk about the proposed multi-agent system for tracking detected objects and then depict future trends on this area. Finally, conclusions are presented.

Previous Efforts on Automated Surveillance Systems

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

Video Surveillance and Monitoring (VSAM)

An early research effort made to address this problem was a system proposed by Collins, Lipton, Fujiyoshi, and Kanade (2001) at Carnegie Mellon University entitled "Video Surveillance and Monitoring" (VSAM). The architecture of VSAM consists of an operator central unit, several sensor processing units, a graphical user interface (GUI), and several visualization nodes.

The sensor processing units perform object detection by means of adaptive background subtraction and three-frame differencing. This information is sent to the operator central unit. That is, the operator control unit receives the processed data from the entire sensor processing units distributed over the area of interest. Once the operator control unit acquires results from the sensor-processing units, 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 the visualization nodes. The user must indicate surveillance tasks to be performed. The operator control unit serves / acts as an arbiter, indicating which unit must perform a given task. The tasks include tracking a given object, displaying views fields, and / or creating a region of interest.

This system has advantageous characteristics such as providing several points of visualization through the network, saving bandwidth by sending symbolic information about the detected object. However, the system still needs human supervision. The tracking of moving objects is only performed when the user assigns the task. The operator control unit acts as an arbiter. Therefore, tasks cannot be completed if it fails. The architecture detects and classifies objects present on the scene and shows all this information in a report to the user. Then, the system waits for the user to assign new tasks.

Distributed Intelligence for Visual Surveillance

Framework based on cooperative agents for visual surveillance was proposed by Remagino, Shihab, and Jones (2004) at Kingston University. The system consists of several cameras distributed over the area of interest where the view field of the cameras can be overlapped. The framework defines three agents: ground plane agent, camera agent, and object agent. 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 three-dimensional (3D) trajectory of the object and extracts the set of sub-images containing the event in 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 a moving or static one. However, it is not clear whether this detection algorithm will discard shadows cast by a moving object or if the algorithm is robust against changes of illumination. Several cameras may instantiate a number of agents tracking the same event, which may result in waste of resources until the agents merge. Also, the camera agent is responsible for detecting the moving objects, extracting parameter models, and assigning objects to the object agents, while the object agent just stores the values of parameters. Therefore, the burden of processing resides only on the camera agent.

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, Rosario, and Pentland. (2000) at MIT. Visual surveillance requires understanding how humans behave and interact among them to achieve a secure area. This approach models the person-to-person interaction using statistic learning techniques known as hidden Markov (Rabiner, 1986, 1989) and coupled hidden Markov models (Brand, 1997) to teach the system to recognize the 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 up by averaging N previous frames. This technique is known as eigenbackground subtraction. The 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.

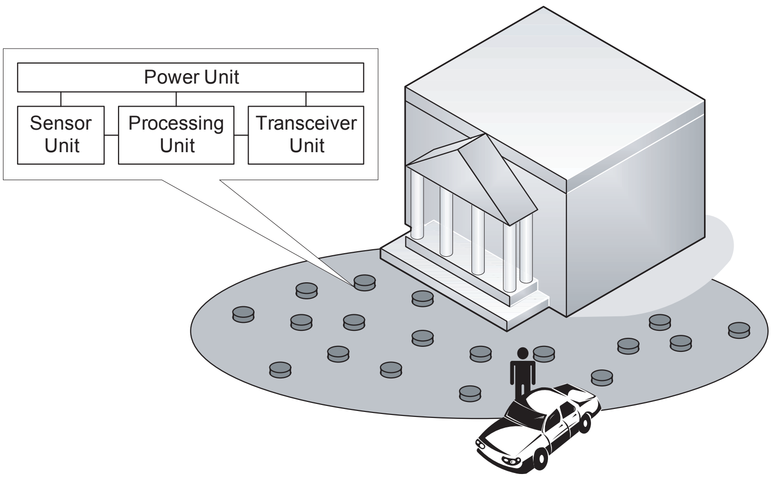
All those previous approaches fail to establish a trade-off between complexity, accuracy, and automation. Surveillance systems require automated detection and tracking that can be implemented in real-time while consuming a moderate amount of resources. The information that can be shared through the network must be limited to relevant data. Moreover, the communication must be restricted to nodes that share the same monitoring area giving the opportunity of discarding redundant information and saving energy.

Basis for the Proposed Development Approach: Distributed Sensor Network

Automated surveillance systems must perform three basic tasks: *object detection, tracking,* and *threat analysis*. The proposed surveillance system consists of a network of cameras spatially distributed, therefore, object detection and tracking requires distributed computing and cooperation among the nodes to perform their tasks. In order to perform threat analysis, a high-level description of the area under surveillance is also provided. The proposed framework can provide distributed computing, cooperation among nodes, and a high-level description of the monitored area. In essence, the backbone of the proposed system is a *distributed sensor network*.

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 with 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 a *data fusion system*. The resulting data is then used to infer the state of the monitoring area. Since there are several hundreds of sensors distributed over an area, multiple instantiations of the same event is possible. In order to avoid transmission of redundant information of the same event, sensors must cooperate to detect such instantiations and merge those different datasets into one single set. There are several applications that can benefit from the proposed framework such as surveillance of battlefields, intrusion detection, improvement of intelligibility under noisy conditions, automatic control of air-conditioners and humidifiers in buildings, environmental monitoring, and remote health monitoring. Several applications are discussed by Iyengar (2005).

Figure 2. A sensor node in the proposed system



The *wireless sensor network* must comply with several characteristics such as intelligence in each node implemented by a processing unit, the performance must not degrade because of the spatial distribution, the network must be able to accommodate diverse sensors, and the network must operate even if some nodes have failed. Data processing must be done in real-time, because sensor data are time-sensitive. In summary, distributed sensor networks must be intelligent, reliable, and robust. In sensor networks, the data is stored and retrieved from several nodes spatially distributed by means of the query processing system as explained by Woo (2004).

Each sensor node is a tiny embedded device composed of four basic units: sensor, processing, communication and power unit, illustrated in Figure 2. The sensor unit may include any of the following sensors: seismic, acoustic, radar, infrared, humidity, light, temperature, pressure, vibration, and radioactivity sensors. The processing unit is composed of a digital signal processor (DSP) or ARM processor capable of performing pre-processing on the collect data. The communication unit is formed for a transceiver and gives the nodes the capability to communicate among them to achieve their goal. Finally, the power unit is in charge of supplying voltage for the entire node operation. Changing batteries for sensors in short intervals of time is infeasible due to the large amount of nodes. Therefore, durable sources of energy have been studied over the years (Meninger, 2001; Lal, 2004). Also, several techniques for efficient use of the energy inside the node have also been studied (Wentzloff, 2004; Sinha, 2001).

Proposed Approach for Automated Object Detection and Tracking

A hierarchical distributed sensor network approach is proposed to enable automated object detection and tracking. In the proposed approach, there are two units: object processing unit (OPU) and scene processing unit (SPU).

OPU is an embedded system capable of detecting moving object in the area under surveillance. It contains an image sensor to provide raw video data to be processed by a background subtraction system to obtain a foreground mask. The *foreground mask* contains moving objects, appearing and disappearing objects while discarding changes due to illumination variations. Once an object has been detected, the information is passed to a multi-agent tracking system. The multi-agent tracking system extracts object model parameter values and stores them until the time to send them to SPU.

SPU is a high-level analysis and storage node. It receives information from several OPUs distributed throughout the area of interest. The information that SPU receives consist of object model values, object segments, and the tracking history. Then, SPU classifies each object into predetermined sets. The sets are broadly organized into human-individuals, animals, and vehicles. Once this information is obtained, it is used to analyze if a threat pattern exists.

OPUs are organized into clusters to ensure high correlation between data collected for nodes within a group. A cluster-head is elected for each cluster. The cluster-head is in charge of

Figure 3. Overall proposed architecture

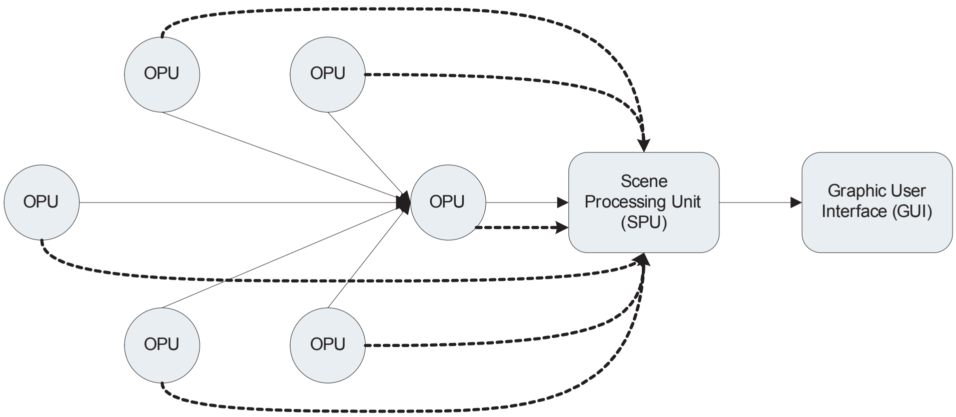
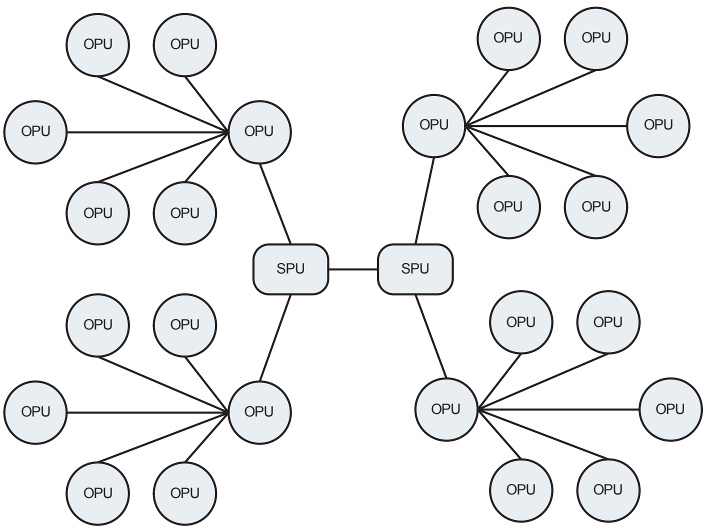


Figure 4. Connectivity of the whole system (all connections are wireless)



merging cluster data before sending them to SPUs. Therefore, cluster-heads function as a gateway between the cluster and the SPU. The overall organization of the proposed approach is illustrated in Figure 3. Each SPU includes a graphic user interface (GUI), to provide several visualization points through the network. It is noted that an SPU can be connected to more than one cluster as illustrated in Figure 4. The SPUs and OPUs have independent functionalities. However, a meaningful cooperation can be achieved through coordination between them. In order to track moving objects, cooperation is needed between OPUs. When an object moves out of the area monitored by an OPU, then the information is sent to the next corresponding OPU the previous OPU does not suspend detection and tracking. SPU analyzes further the information sent by the OPUs. Also, it is in charge of requests information to the OPUs such as video footage.

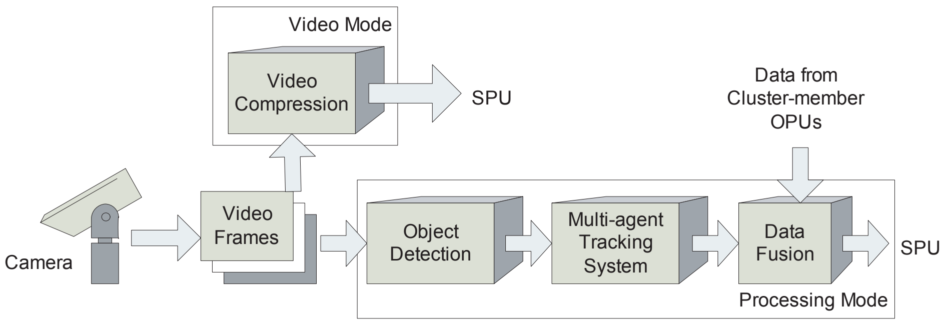
The information that flows through the network is only the object parameter values, that is, no video is transmitted from the OPUs to SPU. This way, the use of bandwidth is optimal. However, the architecture allows the possibility of viewing the actual video footage on the cameras. At a given time the user can indicate by means of the GUI his or her desire of watching the video footage 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 processing mode* will detect and track objects and send only the object model information. The second mode is *video mode* in which the OPU sends compressed video to the SPU.

Object Processing Unit

OPU is a low level sensor node consisting of an embedded system containing an image sensor. OPU obtains raw video stream processed by a background subtraction technique known as *Wronskian change detector*. Then, a foreground mask of the scene under surveillance is sent to the *multi-agent tracking system*. Agents are in charge of obtaining the *object model* parameters. Object model consists of the velocity, acceleration, and direction of movements as well as segments of the image containing the detected objects. Once an object has been detected, the agent system initiates a tracking file for each object and records the object model information. This information must be stored until the transmission to the SPU takes place. OPU functions are decomposed into video and processing mode. *Video mode* takes raw video stream and compresses it using *discrete cosine transform* (DCT). *Processing mode* performs the functions already described here. Processing mode is the default operation mode. Figure 5 illustrates the functional diagram of the OPU.

SPU requires a high-level interpretation of the data acquired by the OPUs rather than individual data of each cell; therefore, the data aggregation is performed before sending the information. OPUs are connected into static clusters. Clusters are created beforehand to ensure the maximum correlation. The data aggregation technique is used to combine the

Figure 5. General functions performed at OPU



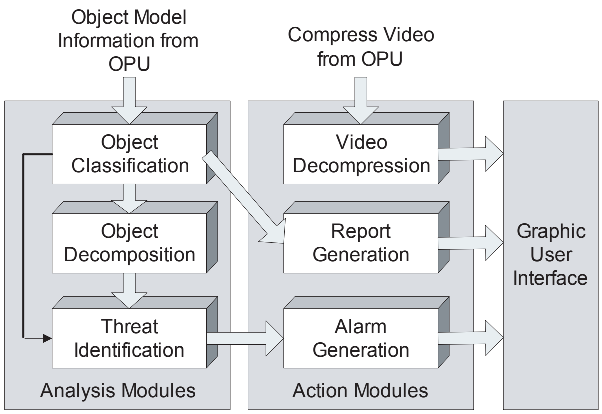
object model information into a smaller set that retains all effective data. In each cluster, one unit acts as a cluster-head. A cluster-head is responsible for data aggregation and information transmission to SPU. Also, a cluster-head acts as a gateway between a cluster and a SPU. Since the cluster-head has more activities than the cluster-members, being a cluster-head is more energy-intensive than being a cluster-member only. A rotation of the cluster-head position is done in order to maximize lifetime of each node. The information is sent to the SPU in rounds; each round begins with a cluster-head selection. The duration of each round is kT, where T is a fixed interval of time and k is a positive integer. After the cluster-head selection, the aggregation and transmission of information is done. The selection of cluster-head is realized by using a modified LEACH cluster head selection algorithm provided in (Heinzelman, 2002).

Cluster members retain the information sent to the cluster-head until they receive a successful transmission message. Then, the information will be discarded in order to have enough space for new information. The cluster members wait for the successful transmission message from the cluster-head for a limited interval of time. While waiting, the cluster members still perform their actions. If they do not receive the message within this interval, a new cluster-head will be elected and information will be re-sent.

Scene Processing Unit

SPU is a high-level analysis and storage node dedicated to perform threat identification based on the information sent by multiple OPUs. SPU has a broader look at the scene under observation because it receives the information from several OPUs. Thus, the SPU can perform accurate detection of suspicious activities. Along with object model and tracking information, a segment of the image containing the detected object is received by the SPU. This segment could be used by the classification and decomposition procedure. The object could be classified into person, animal, or vehicle sets. A person set contains a single person or a group of persons, while the vehicle set contains sedan, truck, delivery truck, or 18

Figure 6. Functional diagram of SPU tasks



wheelers. If the detected object has been classified into person set, then the decomposition procedure will take place and a person will be 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. The information will be sent to the threat detection procedure. Based on that information the SPU will decide what action will be taken, such as taking a closer look on the scene or an 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 they require high computation capabilities and memory as well as a database of threat patterns. SPU functions are classified into analysis and action modules. Analysis modules contain object classification, object decomposition, and threat identification, while video decompression, report generation, and alarm generation are action modules.

Graphic User Interface

The graphic user interface (GUI) provides a daily report of actions in the area of interest. The report includes all information collected from OPUs on the particular object observed. It also provides a tool to observe video from the cameras. The video information is sent compressed to the SPU, which is in charge of decompressing the video and showing it by means of GUI.

The Proposed Multi-Agent Object Detection and Tracking System

A smart tracking system requires intelligence to perform its activities, which a *multi-agent system* can provide. Since the early 1990s, agent paradigms have been the subject of intense study. *Agents* are an extension of the object-oriented paradigm and have been suggested as a breakthrough of computer science (Jennings, 1998). 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. Agent-based systems 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, and problems whose solutions are drawn from distributed autonomous experts, for example, healthcare providers as suggested by Nwana (1999) and Dietl (2001).

Agents can effectively address several applications, such as mobile computing environment on database systems (Pissinou, 1997). This architecture employs agents as a representative of mobile system. Agents can provide an analysis tool for team behavior (Nair, 2004). Agents have also been employed in marketplace modeling. One of the major research topics in these applications is the negotiation protocol. The 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 Palaniappan (1992). 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 Hodge (2003).

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

- It provides a 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 is temporally and spatially distributed. Cooperating agents that collect spatial and temporal information through the entire area are used to solve the problem.

Our approach to scene understanding incorporates agents under the following scheme. The area of interest is divided into several sub-areas in agreement with camera range view as illustrated in Figure 6. Each region corresponds to a sub-area where the camera has the best view. Each sub-area has assigned a camera and a *region agent*. A fixed video camera with a wide range of view delivers the raw video stream to be analyzed. The object detection block delivers a foreground mask containing only the event of interest. This foreground mask is sent to the region agent.

The region agent segments the image using the foreground mask. Each segment is sent to the *object agents* that have been already spawned by the region agent. If any object agent does not recognize a segment, then a new agent is spawned to track that object. The object agent is responsible for updating the object model based on information subtracted from

Figure 7. Segmentation of the area of interest for two different camera dispositions

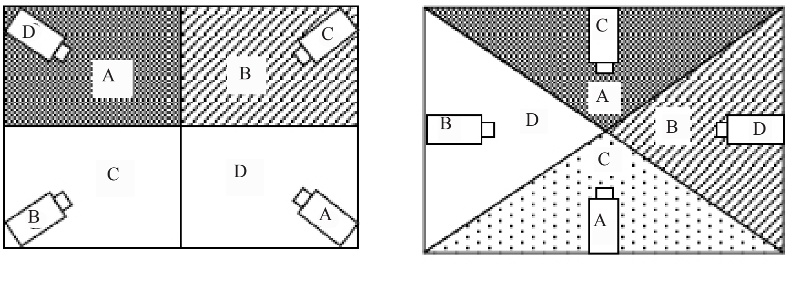
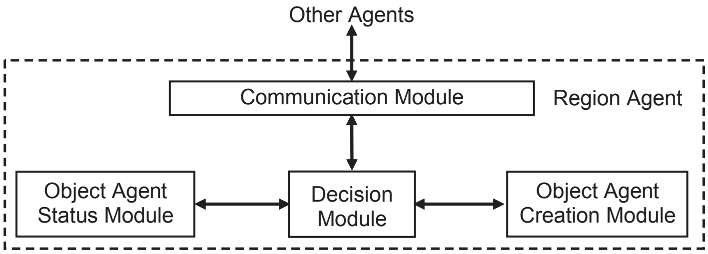


Figure 8. Region agents model



several frames. When an object approaches the border of the area monitored by the region agent, this agent must communicate with the proper agent to send all the information on the object to it. The region agent negotiates proper handoff of moving objects leaving its area with its neighbor.

Each object agent has its *tracking database* to store all the values of the model parameters. The task of the object agent is to identify its assigned object and update image segment as well as trajectory in its tracking database, while the region agent is responsible for creation of object agents and assignment of detected object.

Region Agent Model

The region agent is responsible for monitoring its area as well as coordinating object agents already assigned to the detected objects. In order to perform its activities, a region agent consists of four modules: communication module, object agent status module, object agent creation module, and decision module. Figure 7 describes the agent model. When a new event has been sensed, the region agent must create a new object agent responsible for tracking and updating the object model of the recently detected event.

Object Agent Status Module

The region agent (RA) functions as a coordinator for all the object agents (OAs) that have been created by it. In order to perform its activities, the RA must know the status of the OA. When a new frame has arrived, the RA is responsible for segmenting the frame. Each segment of the frame contains a detected object. The RA marks all the OAs as *not-identified* to indicate that none of the OAs has identified the object present at the scene. When an OA recognizes its object, it will send an acknowledgment message to the RA, then update its status as *positive-identification*.

Decision Module

The decision module is in charge of generating all messages for the other agents. When a new frame arrives, the OA status must be updated and messages are sent to OAs announcing the arrival 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.

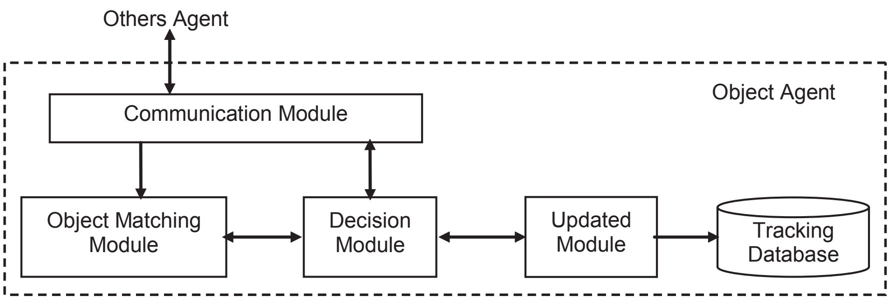
Communication Module

The communication module allows the RA to exchange information with the other agents via a predefined set of messages. The decision module chooses the types of messages and their contents. The agents in the proposed system utilize a protocol based on the Knowledge Query Manipulation Language (KQML) proposed by Finin (1992). KQML is based on *speech act theory* (Searle, 1970) and is a popular protocol that is being used widely for communication among agents (Huhns, 1998; Weiss, 1999). All the necessary information for the correct interpretation of the message is included in the transmission.

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 informing the RA that a positive match was established. To execute its task, the OA uses communication, object matching, and decision modules. 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 8. The update module is responsible for updating the new object parameter values in the tracking database. The decision module generates the message to communicate with the RA. The OA must inform when a positive match has been established. Also, the decision module chooses when the update process must be performed.

Figure 9. Region agents model



The object matching module recognizes if the segment contains the assigned object. The decision is made based on the Mahalanobis distance. The Mahalanobis distance is a technique to determine similarity between a set of values and an unknown sample (Kamei, 2002). The Mahalanobis distance takes the information of the variance and covariance between variables. This 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 using Equation 1, where I represents 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.

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

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

Table 1. Fact definition

Fact	Attributes	Default	Meaning
Image_Segment	Segment: Matrix containing a image segment	Null	Segment containing the object to be match
Position_on_Frame	Coordinates x, y: integer	(0,0)	Position coordinates of the upper left corner of the segment
Distance_to_Border	Distance: real	Max	Distance of the object to the border
Approaching_Border	Approach: Boolean	False	Flag for object approaching border area
Velocity	Velocity: real	0	Velocity of the detected object
Acceleration	Acceleration: real	0	Acceleration of the detected object
Heading	Heading: integer	0	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	Null	Name of the Object Agent
RegionAgentID	Name: string	Null	Name of the Region Agent

multi-agent application. 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 their activities. RA publishes image segments and each OA takes an image segment and identifies if it 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 among each other to cooperatively solve a designated problem. The communication is performed via messages. Each message contains parameters that possess 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 1 presents the set of facts defined for this application.

Object Detection through Background Subtraction Technique

Change detection plays a key role in real-time image analysis. Detection on the scene under observation includes moving, adding, or removing objects. The detection procedure must be able to differentiate between noise or illumination variations and actual movement. One key issue is robustness against illumination changes. A review of several approaches has been studied by Cheung (2004). The most instinctive technique is *frame differencing,* followed by thresholding. Change is detected if the difference of the corresponding pixels exceeds a preset threshold. The advantage of this technique is its low computational complexity. However, it is very susceptible to noise and illumination changes.

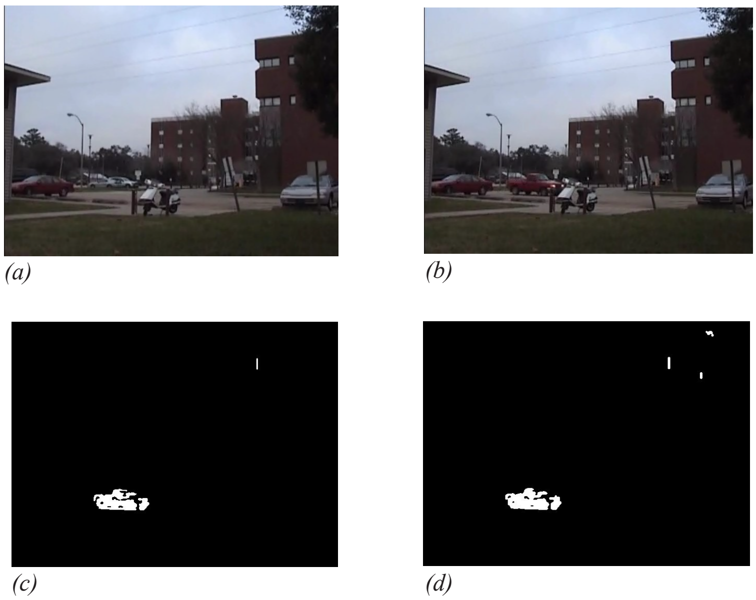
Median filter is one of the most popular background subtraction techniques (Cucchiara, 2003). The median of each pixel of all the frames in the buffer constitutes the background estimation. Background pixels are considered to be those that stay on more than half of the frames on the buffer. However, this technique requires a buffer large enough to store L frames. *Recursive background subtraction techniques* do not require a buffer of previous frames. They recursively update the background model based on each input frame. Any error in the background estimation can remain for a long period of time due to its recursive nature. The most popular recursive technique is *Mixture of Gaussian (MoG)* (Friedman, 1997). This method models each background pixel by a mixture of K Gaussian distributions (K is a number between 3 and 5). Different Gaussians are assumed to represent different colors. The weight parameter of the mixture represents the time proportions that those colors stay in the scene. The probable background colors are the ones that stay longer and more static. However, the technique is computationally intensive; its parameters require careful tuning, and it is very sensitive to sudden changes in global illumination.

Wronskian change detector (WCD) employs the Wronskian of intensity ratios as a measure of change (Durucan, 2001). A large mean or large variance of the intensity ratios increases the Wronskian value. This method can detect object interiors and structural changes. Also, WCD is robust against illumination changes. WCD is a suitable algorithm to be implemented

Table 2. Background subtraction techniques

Method	Adaptability	Precision	Complexity	Tuning	Global Illumination Changes	Storage Requirement
Frame Differencing	High	Low	Low	Simple	Sensitive	1 frame
Median Filter	High	Medium	Medium	Simple	Less sensitive	L frames
Mixture of Gaussian	Low	High	High	Complex	Sensitive	None
Wronskian Change Detector	High	Medium	Medium	Simple	Robust	1 frame

Figure 10. 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



in real-time due to its low complexity. Also, this technique requires only one previous frame; therefore, we adopt it for application in which resources are limited. A comparison of the discussed methods is presented in Table 2. WCD offers a tradeoff between complexity and storage requirements while achieving medium precision and robustness against global illumination changes.

WCD is a non-recursive background subtraction technique that distinguishes changes based on intensity values. Consequently, WCD requires conversion of images into luminance values. Changes are detected based on intensity ratios variance. In order to determine if a change has occurred, a region of support is assigned to each pixel. The size of the region of support can vary from 3 × 3, 5 × 5, and 9 × 9 pixels. The center pixel of the region of support is replaced by the vector formed with the pixel and its neighbors.

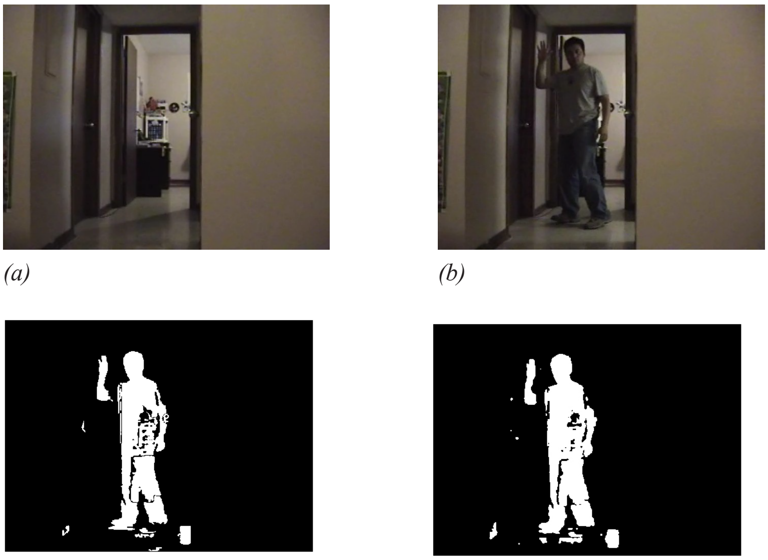
If two vectors are linearly independent, then a change can be assumed. A simple and rigorous test for determining the linear dependency of vectors is the Wronskian determinant. WCD exploits the fact that the ratio of luminance values from two light sources helps to quantify the difference between the light sources. The WCD is computed by the Equation 2, where x_i and y_i are the components of the vector for the current frame and the previous frame respectively and n is the dimension of the vector. TH represents the value of the threshold.

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

$W(x/y)$ detects changes corresponding to dark zones, while its inverse ratio $W^*(x/y)$ finds if a change has occurred in bright zones. Therefore, computing both values allows robust detection against global illumination changes.

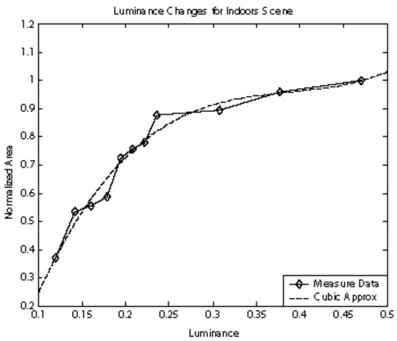
Simulation results show that regions of support larger than 3 × 3 do not provide better results but require more resources. Therefore, in our approach a fixed 9-dimension vector has

Figure 11. 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 wtih a 5×5 window

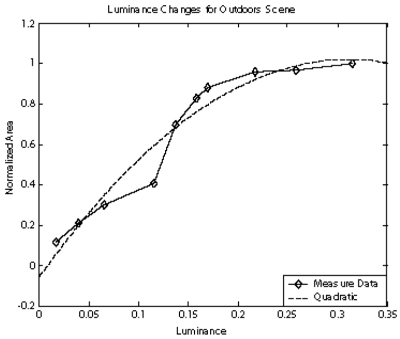


(a) (b) (c) (d)

Figure 12. Detection performance for different luminance values, (a) results for indoor scenes; (b) results for outdoor scenes



(a)



(b)

been selected. 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 value. The resulting foreground mask has zero values for elements on the background and 255 values for detecting moving objects. Figure 9 shows result for an outdoors image using window size of 3 × 3 and 5 × 5. Figure 10 illustrates simulation results for background subtraction of an indoor image. The results for two different size windows are almost the same in these cases.

The background subtraction algorithm must be robust against the 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 11 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. The 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.

Future Trends

Threat recognitions must be done by analyzing behaviors of detected objects. There are two main sources of attack: humans and vehicles. 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 the hidden Markov model can be employed to detect the criminal pattern activity. However, the nature of these events imposes several challenges such as small and imbalanced training sets. In view of the fact that criminal activity is a rare event, the amount of training data is small. Furthermore, it is desirable that new criminal patterns 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 into generative or discriminative ones. The generative models such as the 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, the discriminative models estimate the posterior for each class without modeling the class density. Solutions to these problems have been proposed in Maciel (2000) and Heisele (2003). The coupled hidden Markov and hidden Markov models have been applied toward behavioral analysis and agent have been employed to provide training set for these models (Meninger, 2001).

Another issue to consider is security in the sensor network. Since nodes are limited in energy, computation, and communication capabilities and interact with their physical environment and people, they are prompt to a variety of attacks. These attacks include node capture, physical tampering, and denial of service. Data encryption and access control is one approach to guarantee security and privacy.

Current public-key cryptographic primitives are expensive in terms of computational complexity; therefore, it is not suitable to employ such technique in computational-limited nodes. A new random-key predistribution scheme that is resilient against node compromise is needed as well as the hardware support for such approach (Dimitrijevic, 2003). Denial of service attack disrupts the network's operation by broadcasting a high-energy signal. If the transmission is strong enough, the entire network communication could be jammed. A defense against jamming is to employ spread-spectrum or frequency hopping communication scheme. Therefore, a cryptographically secure spread-spectrum of frequency hopping radio is greatly needed.

An essential service for sensor network is routing and data forwarding. However, routing protocols are susceptible to node-capture attacks. A big step toward a secure network will be achieved by creating secure routing protocols. Sensor network require solutions that are fully distributed and incurred in low cost in terms of communication, power, and memory resources. Several issues related to security and privacy on sensor network are presented by Perrig (2004) and Chang (2003).

Conclusion

A distributed scene understanding architecture has been presented. The architecture consists of a hierarchical sensor network comprising two units: object processing unit and scene processing unit.

OPU is a low-level embedded system containing an image sensor that provides a video stream used to automatically detect and track object present on the scene. Object detection is performed by means of Wronskian change detector. WCD delivers a foreground mask containing detected objects. This foreground mask is employed by the multi-agent system to track each object. Multi-agent system extracts object model values and stores it until they are transmitted to SPU.

SPU is a high-level analysis and storage node dedicated to identifying threat patterns. In order to perform its task, detected objects are classified into predefined sets. Then, the objects are decomposed and the information is supplied to a threat identification block along with the tracking information. If a threat is identified then an appropriate action will be taken. SPU continuously generates a report of activities containing the detected object and tracking information.

The proposed architecture represents an effort toward an intelligent surveillance system capable of automatically detecting suspicious activity using wireless sensor networks. Continued research in this direction will allow preventing catastrophic events. The more intelligence is incorporated into a surveillance system, the closer we will achieve a secure environment. Ethical issues related to this technology have to be addressed by a mixture of technological and social efforts.

Acknowledgments

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, as well as the support of the National Science Foundation NSF, INF 6-001-006.

References

Brand, M., Oliver, N., & Pentland, A. (1997). Coupled hidden Markov models for complex action recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, (pp. 994-999).

Chan, H., & Perrig, A. (2003). Security and privacy in sensor network. *IEEE Computer, 36*(10), 103-105.

Cheung, S-C. S., & Kamath, C. (2004, January). Robust techniques for background subtraction in urban traffic video. In *Proceedings on Video Communication and Image Processing SPIE Electronic Imaging*, San Jose, California, UCRL-JC-153846-ABS, UCRL-CONF-200706.

Collins, R. T., Lipton, A. J., Fujiyoshi, H., & Kanade, T. (2001, October) Algorithm for cooperative multisensor surveillance. In *Proceedings of IEEE, 89*(10), 1456-1477.

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

Cucchiara, R., Piccardi, M., & Prati, A. (2003). Detecting moving objects, ghost and shadows in video stream. *IEEE Transaction on Pattern Analysis and Machine Intelligent, 25*(10), 1337-1342.

Culler, D. E., & Hong, W. (2004). Wireless sensor networks. *Communication of the ACM, 47*(6), 30-33.

Dick, A. R., & Brooks, M. J. (2003, December). Issues in automated visual surveillance. In *Proceeding of 7th Digital Image Computing: Technique and Applications*, Sydney, Australia, (pp. 195-204).

Dietl, M., Gutmann, J. S., & Nebel, B. (2001, November). Cooperative sensing in dynamic environments. In *Proceeding of International Conference on Intelligent Robots and Systems*, Hawaii, 3, (pp. 1706-1713).

Dimitrijevic, Z., Wu, G., & Chang, E. Y. (2003, December). SFINX: A multi-sensor fusion and mining system. In *Proceedings of the IEEE Pacific-rim Conference on Multimedia. Singapore*, (pp. 15-18).

Durucan, E., & Ebrahimi, T. (2001). Change detection and background extraction by linear algebra. *Proceedings of the IEEE, 89*(10), 1368-1381.

Finin, T., McKay, D., & Fritzon, R. (1992). *An overview on KQML: A knowledge query and manipulation language* (Technical Report No.CS-94-12). Maryland: University of Maryland, Computer Science Department.

Friedman, N., & Russell, S. (1997). Image segmentation in video sequences: A probabilistic approach. *Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence*, (pp. 175-181). Morgan Kaufmann Publisher Incorporated.

Green, M. W. (1999). *The appropriate and effective use of security technologies in U.S. schools. A guide for schools and law enforcement agencies* (Tech. Rep. NCJ178265). Sandia National Laboratories.

Hampapur, A., Brown, L., Connel, J., Pankanti, S., Senior, A., & Tian, Y. (2003, December). Smart surveillance: Application, technologies and implications. In *Proceeding of ICICS-PCM*, (pp. 1133-1138).

Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol for wireless microsensor networks. *IEEE Transaction on Wireless Communication, 1*(4), 660-670.

Heisele, B. (2003). Visual object recognition with supervised learning. *IEEE Intelligent Systems, 18*(3), 38-42.

Hill, J., Horton, M., Kling, R., & Krishnamurthy, L. (2004). The platforms enabling wireless sensor networks. *Communication of the ACM, 47*(6), 41-45.

Hodge, L., & Kamel, M. (2003). An agent-based approach to multisensor coordination. *IEEE Transactions on Systems, Man, And Cybernetics — Part A: Systems and Humans, 33*(5), 648-662.

Huhns, M. N., & Singh, M. P. (1998). *Readings in agents*. San Francisco, CA: Morgan Kaufmann.

Iyengar, S. S., & Brooks, R. R. (2005). *Distributed sensor networks.* Boca Raton, FL: Chapman & Hall/CRC Press Company.

Jennings, N., & Wooldridge, M. (1998). Applications of intelligent agents. *Agent technology: foundations, applications, and markets* (pp. 3-28). Berlin, Germany: Springer-Verlag.

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

Lal, A., & Blanchard, J. (2004). The daintiest dynamos. *IEEE Spectrum, 41*(9), 36-41.

Maciel, B. D., & Peters, R. A. (2000) A comparison of neural and statistical techniques in object recognition. *Proceeding of IEEE International Conference on Systems, Man and Cybernetics, 4*, 2833-2838.

Meninger, S., Mur-Miranda, J. O., Amirtharajah, R., Chandrakasan, A. P., & Lang, J. (2001). Vibration-to-electric energy conversion. *IEEE Transactions on VLSI Systems, 9*(1), 64-76.

Nair, R., Tambe, M., Marsella, S., & Raines T. (2004). Automated assistant for analyzing team behaviors. *Autonomous Agents and Multi-agents Systems Journal, 8*(1), 69-111.

Nwana, H. S., & Ndumu, D. T. (1999). A perspective on software agents research. *The Knowledge Engineering Review, 14*(2), 1-18.

Oliver, N. M., Rosario, B., & Pentland, A. P. (2000). Bayesian computer vision system for modeling human interactions. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *22*(8), 831-843.

Palaniappan, M., Yankelovich, N., Fitzmaurice, G., Loomis, A., Haan B., Coombs, J. et al. (1992). The envoy framework: An open architecture for agents. *ACM Transaction on Information Systems, 10*(3), 233-264.

Perrig, A., Stankovic, J., & Wagner, D. (2004). Security in wireless sensor networks. *Communication of the ACM, 47*(6), 53-57.

Pissinou, N., Makki, K., Hong, M., Ji, L., & Kumar, A. (1997, November). An agent based mobile system. In *Proceeding of 16th International Conference on Conceptual Modeling*, Los Angeles, CA (pp. 361-374).

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. *IEEE Acoustic Speech and Signal Processing Magazine, 3*(1), 4-16.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257-286.

Remagnino, P., Shihab, A. I., & Jones, G. A. (2004). Distributed intelligence for a multi-agent visual surveillance. *The Journal of the Pattern Recognition Society, 37*(4), 675-689.

Searle, J. R. (1970). *Speech acts: An essay in the philosophy of language*. Cambridge University Press.

Sinha, A., & Chandrakasan, A. (2001). Dynamic power management in wireless sensor networks. *IEEE Design and Test of Computer, 18*(2), 62-74.

Vieira, M. A. M., Coelho, C. N., da Silva, D. C., & da Mata J. M. (2003, September). Survey on wireless sensor network devices. In *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, Lisbon, Portugal (pp. 537-544).

Weiss, G. (1999) *Multiagent system: A modern approach to distributed artificial intelligence*. Cambridge, MA: Massachusetts Institute of Technology Press.

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

Woo, A., Madden, S., & Govindan, R. (2004). Networking support for query processing in sensor networks. *Communication of the ACM, 47*(6), 47-52.