

Autonomous Decentralized Systems based Approach to Object Detection in Sensor Clusters*

Ruth AGUILAR-PONCE[†], Ashok KUMAR[†], J.Luis TECPANECATL-XIHUITL[†],
and Magdy BAYOUMI[†], *Nonmembers*

SUMMARY This work deploys Autonomous Decentralized System (ADS) based formulation to cluster of networked visual sensors. The goal is to utilize and integrate the sensing and networking capabilities of the sensors with the systematic and autonomous features of ADS to perform visual surveillance through object detection in the covered areas of interest. In the proposed approach, several cells are distributed through an area of interest called *Autonomous Observer Cell*. The decentralized subsystems detect and track moving objects present on the scene by looking through a camera embedded in each sensor. These subsystems form a cluster and each cluster sends information to an *Autonomous Analysis Cell* that determines if an object of interest is present. The Autonomous Observer Cells share a common data field and a cluster-head works as a gateway between the cluster and the Autonomous Analysis Cell.

key words: *Autonomous decentralized system, multi-agent system, cluster network.*

1. Introduction

Autonomous Decentralized System, proposed by Dr. Mori [1], is inspired from the concepts of living organisms. A living organism is composed of cells; each cell is completely independent from one another and performs its own function in unison with the other organs of the body. Similarly, ADS was proposed as a collection of several subsystems that cooperate among them to perform their desired function. The system is defined under two assumptions [1]: i) faulty parts are normal, and ii) the system is a result of integration of subsystems.

Autonomous Decentralized System must comply with three characteristics: on-line expansion, fault tolerance, and on-line maintenance. These requirements decrease the turn down time in any large production system. On-line expansion assures that when the system is undergoing an update or addition of new capa-

bilities, no shutdown will be necessary. ADS also provides on-line testing to assure that every component of the systems is working properly. To ensure that the requirements are met, the subsystems must be equal, self-contained and locally manageable. ADS has been extensively used in computer-aided automated manufacturing [2], [3] network services applications and process control such as Tokyo Metropolitan Train Traffic Control [4], [5].

We propose the use of autonomous decentralized system for surveillance task. Surveillance systems require on-line expansion, and it is highly desirable to expand the surveillance network without shutting the existing network. Existing surveillance systems typically consist of several cameras distributed through an area and connected to monitors in a central operator room, where human personnel are in charge of reviewing and detecting suspicious activity. Despite the technological advances individually made for networking and computing capabilities, there are challenges to overcome before a reliable automated surveillance system is realized [6]. These technical challenges include system architecture design, object identification, tracking and analysis, restrictions on network bandwidth, physical placement of cameras and installation cost.

An early research effort to address this problem was a system proposed by Carnegie Mellon University - Video Surveillance and Monitoring (VSAM) [7]. The VSAM architecture consisted of a Central Operator Unit that coordinated and assigned task to Sensor Processing Unit. The Sensor Processing Unit analyzed the video to detect entities or events and transmitted this information to the Central Operator Unit - which took the information and showed the report to a guard through a graphic user interface and waited for the person in charge to assign task. This system needed human intervention to decide what action must be performed.

A framework based on cooperative agents for visual surveillance was proposed in [8]. An agent is assigned to each camera. A camera agent is responsible for detecting and tracking objects in their range of view. Also it is responsible to create an object agent in charge of identifying the event and describing its activity. Object detection is done by comparing with an adaptive reference image. This method does not guarantee that shadow of moving object or moving trees

Manuscript received March 30, 2005.

Manuscript revised May 30, 2005.

Final manuscript received August 10, 2005.

[†]The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, LA 70504-4330, USA

*The authors acknowledge the support of the U.S. DoE, EETAPP program DE97ER12220, the Governor's IT Initiative, DoE award DE-FGO2-04ER46136 and the Louisiana Board of Regents contract DOE/LEQSF (2004-07)-ULL, and the support of NSF, INF 6-001-006. This paper is an extended work of the authors' accepted IEEE-CAMP'05 conference paper in [22].

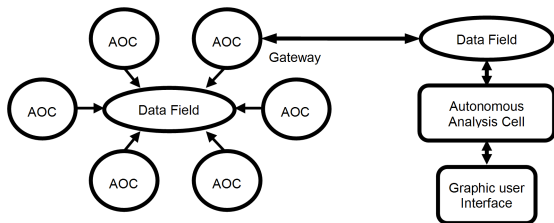


Fig. 1 Overall proposed architecture.

will be discarded, also is weak against global illumination changes. To the best of our knowledge based on literature search, no previous work has been reported using ADS for surveillance in sensor systems.

Our architecture consists of two units: *Autonomous Observer Cell* and *Autonomous Analysis Cell*. The Autonomous Observer Cell is in charge of analyzing the scene captured by its camera, detecting and tracking moving objects. While the Autonomous Analysis Cell will take the information provided by several Autonomous Observer Cells and analyze if a threat pattern is present on the area of interest.

Multi-agent system has been used in combination with autonomous decentralized system to achieve intelligent systems [9]. Example of this combination can be found in several papers [10], [11]. In the proposed architecture, we are employing multi-agent systems to track detected object present on the scene under surveillance. The ADS in our work, provides the enabling framework for system design as well as facilitates communication among the sensor nodes.

The organization of the rest of the paper is as follows. Section 2 describes the proposed architecture and components. Section 3 discusses the design and simulation of the system based on agents, followed by conclusion in Section 4.

2. Proposed Architecture

The proposed architecture monitors an area of interest and identifies object of interest through object detection and tracking. The unit in charge of detection and tracking is called *Autonomous Observer Cell* (AOC). The AOC contains an image sensor, processing unit, transceiver and power unit. The processing unit analyzes the scene captured by the image sensor and detects all moving objects. Once an object has been detected, a tracking history is initiated. In order to determine if the scene contains suspicious activity, detected objects must be classified (e.g. humans or cars). *Autonomous Analysis Cell* (AAC) carries out analysis on detected objects and tracking history. AAC takes information from several AOCs and classifies the objects into humans, group of humans or cars. Once the objects have been identified as member of predefined groups, tracking history is used to determine if the object is acting suspiciously. The overall proposed architecture

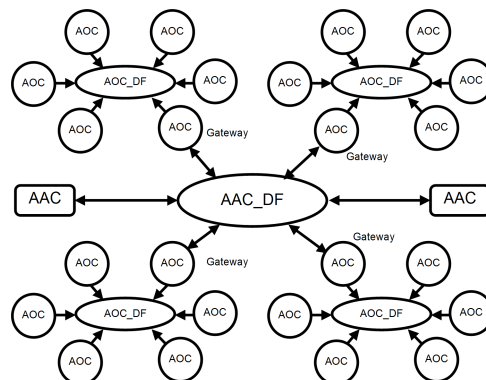


Fig. 2 Connectivity of the whole system.

is illustrated in Figure 1.

The architecture follows the *Data Field Architecture*, where each subsystem is called *Atom* component and all atoms are connected through a common *Data Field* (DF). Each atom is managed by its own processor called *Autonomous Control Processor* (ACP). Communication is performed by broadcasting a message on the DF. Each message contains a *Content Code* (CC) that identifies the data contained in the message.

There are several AOCs distributed through the area of interest and grouped in clusters in such a way that the data in a cluster is highly correlated. AOC are connected through a common DF (AOC_DF) as shown in Figure 2. Each cluster dynamically elects a cluster-head, which integrates all the information collected by the cluster-members. Once the information has been merged, the cluster-head sends the information to the AAC. Therefore, cluster-heads function as a gateway between the cluster and the AAC.

AAC further analyzes detected objects, and also provides visualization points through the network by means of a *Graphic User Interface* (GUI). In order to perform this activity, the AACs must be connected to a DF as illustrated in Figure 2. AAC exchanges information with fellow AACs and AOC through the AAC_DF. The receiver decides which message to accept based on the content code.

The AACs and AOCs have independent functionalities. However, a meaningful cooperation is achieved through coordination between the two. In order to track moving objects, cooperation is needed between AOCs. When an object moves out of the area monitored by an AOC, then information is sent to the next corresponding AOC, however the previous AOC does not suspend detection and tracking. This cooperation is further described in section 3. AAC analyzes further the information sent by the AOCs or requests information such as video footage analyzed by the AOCs.

The information that flows through the network is only the object model information, i.e. no video is transmitted from the AOCs to AAC. This way the use

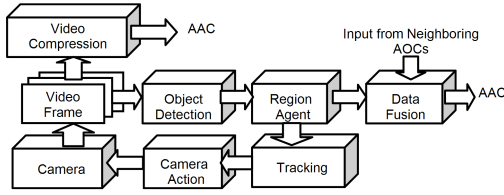


Fig. 3 General functions performed at AOC.

of bandwidth is optimal. However, the architecture allows the possibility of viewing the actual video footage on the cameras. On a given time the user can indicate by means of the GUI his/her desire of watching the video of a given camera. The AOC will then suspend processing and compress the video in order to send it to the AAC. Therefore, each AOC 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 AOC sends compressed video to the AAC.

2.1 Autonomous Observer Cell

The Autonomous Observer Cell (AOC) is in charge of sensing the area of interest and identifying moving objects in the scene. In order to perform its functions, an AOC consists of four basic units: sensor, processing, transceiver and power unit. The AOC 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 data of the sensor.

Figure 3 illustrates the functional diagram of the AOC. Once the image has been obtained, 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 system based on agent is discussed in section 3. The object model information along with the image of the object is sent to the AAC.

2.1.1 Cluster-head Selection

AAC requires a high-level interpretation of the data acquired by the AOCs rather than individual data of each cell. AOCs data are correlated; therefore AOCs are formed into static clusters. Cluster are created beforehand to ensure maximum correlation. Data aggregation technique is used to combine object model information into a smaller set that retains all effective data. In each cluster, one unit acts as a cluster-head. Cluster-head is responsible for data aggregation and information transmission to AAC_DF. Also cluster-head acts as gateway between AOC_DF and AAC_DF. Therefore

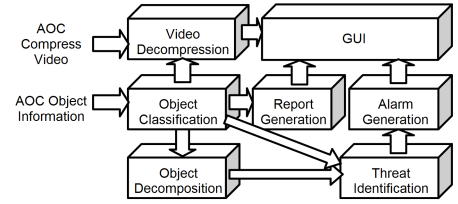


Fig. 4 Functional diagram of AAC tasks.

being a cluster-head is more energy intensive than being a cluster-member only. A rotation of cluster-head position must be done in order to maximize lifetime of each node. The information is sent to the AAC 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 cluster-head selection, aggregation and transmission of information is performed. The selection of cluster-head is realized by using the LEACH cluster head selection algorithm [12].

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

2.2 Autonomous Analysis Cell

Autonomous Analysis Cell has a broader look at the scene under observation because it receives the information from all AOCs in the same area. Thus, the AAC 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 AAC.

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 AAC decides what action will be proper, such as taking a closer look on the scene or an actual alarm detonation. Figure 4 shows a functional diagram of the activities performed at the AAC. The tasks performed by the AAC are computationally complex, thus require

high computation capabilities and memory as well as a database of threat patterns.

2.3 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 AOCs on the particular object observed. It also provides a tool to observe video from the cameras. The video information will be sent compressed to the AAC, and the AAC will be in charge to decompress the video and show it by means of GUI.

2.4 Communication Issues

Mori identifies four different problems that must be solved in order to adopt ADS paradigm on sensor network [13]: direct data broadcasting, data fading, fine data subscription and filtering and finally localized coordination.

- **Direct Data Broadcasting.** Since sensor nodes have limited communication and processing capabilities, data cannot be flooded through the entire network, rather it must be flooded with certain direction. This issue is addressed by forming cluster between the nodes. Each cluster has its own data field, and information is shared only with the nodes within the cluster. Cluster-head functions as a gateway to communicate with other cluster or with the AAC.
- **Data Fading.** When data is flooded through the entire network, fading of data could happen. In order to avoid data fading, data field is partitioned into several data field. Data is broadcast only between nodes in the cluster. Data fusion takes place on the cluster-head, and then is broadcasted to data field of AACs.
- **Fine data subscription and filtering.** In ADS, applications subscribe and filter data based on the content code only. However in sensor network, nodes can perform application-specific data aggregation and caching. Therefore, sensor network may have different filtering techniques. In our architecture, node must form a cluster; each cluster has highly correlated data. Clusters are formed beforehand, to assure correlation between data. Cluster-head performs data aggregation and sends it to AAC_DF.
- **Localized coordination.** ADS requires that each node get all information within entire network to achieve coordination. In sensor network, a single node cannot get all information through the entire network due to its limitations. Therefore localized coordination must be developed. In our architecture, nodes coordinate among nodes within its cluster. Nodes do not require information outside their cluster, achieving localized coordination.

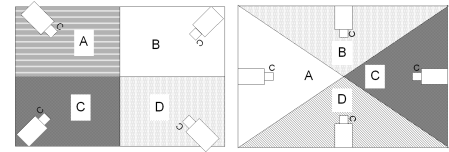


Fig. 5 Division of the area for different camera dispositions.

3. Proposed Cooperative Agents Based Detection and Tracking: Design and Simulation

The agent framework is well suited for application to our scene understanding because it has multiple desirable characteristics, such as mechanism for binding together a set of tasks related to a particular input, clear specification of the interface between these sets and event driven process control [14].

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 solve this 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 5. 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 (RA). Video source is a fixed video camera with wide range of view. Object detection is done by means of background subtraction.

Region Agent 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 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. Object Agent 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 region agent, this agent must communicate with the proper agent to send all the information on the object to it. The Region Agent negotiates with its neighbor proper handoff of moving objects leaving its area

3.1 Region Agent

Region Agent functions as a coordinator for all the object agents that have been created by it. Region agent consists of four modules as illustrated in Figure 6. In order to perform its activities, Region Agent must know the status of the Object Agent. When a new frame has

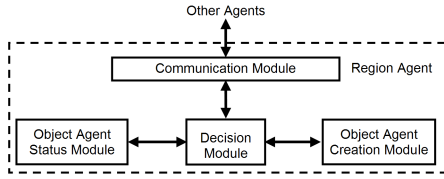


Fig. 6 Region agent model.

arrived, Region Agent is responsible for segmenting the frame. Each segment of the frame contains a detected object. The Region Agent marks all the object agents as no identified to indicate that none of the object agent has been able to identify the object present at the scene. When an object agent recognizes its object, it sends an acknowledgment message to the region agent then updates its status as identified.

Decision module is in charge of generating all messages to the other agents. This module decides the order of transmission to the object agents. When a segment has not been identified for the object agents already created, a new object agent is spawned. When an object is approaching the border of the area, communication with the proper Region Agent is engaged. The decision is based on the heading of the object.

Communication module allows the Region Agent 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. Knowledge Query and Manipulation Language (KQML) [15] is based on speech act theory [16] and is a popular protocol that is being used widely for communication among agents [17], [18]. All the necessary information for the correct interpretation of the message is included in the message itself.

3.2 Object Agent

An Object Agent is responsible for determining if its assigned objects appear on the scene. Also, it is responsible for updating the model and let the region agent know that a positive match was established. To execute its task, the object agent is composed of 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 object agent model is depicted in Figure 7. 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 Region Agent. The Object Agent must inform when a positive match can be established. Also, the Decision Module chooses when the update process must be performed.

The Object Matching Module recognizes if the seg-

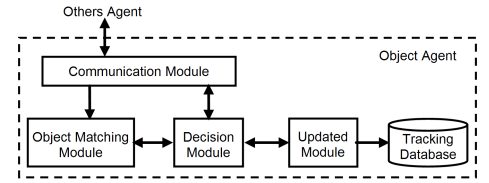


Fig. 7 Object agent Model.

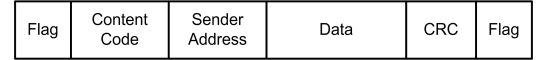


Fig. 8 Message format.

Table 1 Messages exchanged in AOC_DF

CC	Message	Definition
1-12	AGENT	Codes 1 to 12 are reserved for agent communication.
13	TRK	<i>Tracking Information Message</i> contains all tracking information collect for one node.
14	STR	<i>Successful Transmission Message</i> indicates that cluster-head has successfully sent information to AAC_DF.
15	UTR	<i>Unsuccessful Transmission Message</i> denotes a failure on the communicated information to AAC_DF.

ment 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 [19]. 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.

3.3 Content Code Definition

In the DF, each message contains a Content Code (CC) that indicates its meaning. All messages sent to the DF follow the same format illustrated in Figure 8.

The receiver accepts messages based on the content code rather than the sender's address. CC communication allows every subsystem to have autonomy in sending and receiving data because each subsystem does not need to know the relation among sources and destinations. The gateway functions as a bridge between AOC_DF and AAC_DF. Therefore, when a cluster-head receives a message in which content code is of interest to the AOC_DF then broadcast the message into the respective DF and vice versa. Tables 1 and 2 show the CC defined by this application.

In AOC_DF, CCs 1 to 12 are reserved for agent communication. Examples of messages exchanged between agents are *New Frame Arrive* (NFA) and *Agent Next in the List* (ANL), etc. NFA is sent to all object agents that depend on this particular region agent to indicate that a new frame has arrived. The purpose of ANL message is to indicate to which agent the segment of the frame must be sent in case a negative match has

Table 2 Messages exchanged in AAC_DF

CC	Message	Definition
1	MTR	<i>Merge Tracking Information</i> message contains a merge version of the tracking information send by all cluster members. Only the cluster-head can issue this message.
2	RER	<i>Request Report</i> message indicates that the user has requested a report of activities, all AAC must send a report.
3	RVI	<i>Request Video</i> message specifies that AAC has requested a video from some AOC.
4	RGE	<i>Report Generated</i> message contains the report of activities.
5	VRE	<i>Video Requested</i> message contains the video requested by AAC.

Table 3 Fact Definition

Fact	Attributes	Default	Meaning
Image_Seg	Seg: image segment	NULL	Segment containing object to be match
PosFrame	coordinates x, y: int	(0,0)	Upper left object position
BorDist	Dist:real	MAX	Object's distance to the border
AppBor	App:boolean	False	Flag for object approaching border
Velocity	vel:real	0	Velocity of detected object
Accel	acc:real	0	Acceleration of detected object
Heading	head:int	0	Heading of detected object
PosID	ID:boolean	False	Positive match of assigned object
NNegID	NOID:int	0	Amount of negative identification
NSeg	Nseg:boolean	False	There are no more segments to analyze
ObjectAID	name:string	NULL	Name of Object Agent
RegionAID	name:string	NULL	Name of Region Agent

been established.

3.4 Simulation Results

The simulation of the cooperative agents is done by ZEUS toolkit [20]. 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 multi-agent application. We employ the publisher-subscriber model. Region Agent publishes image segments and each Object Agent takes an image segment and identifies if corresponds to its assigned object. If a part of the object cannot be identified by any Object Agent, then the Region Agent creates a new Object Agent 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 3 present the set of fact defined for this application.

Wronskian Change Detector Model (WCD) is used

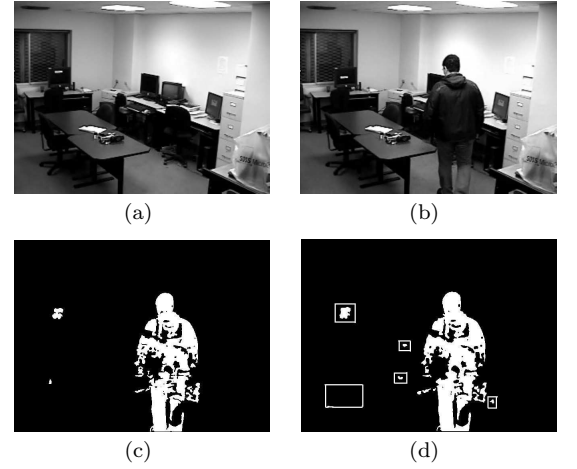


Fig. 9 Object detection results for an indoor scene a. Background image, b. Image containing a moving object, c. Foreground mask using a 3 x 3 window, d. Foreground mask using a 5 x 5 window.

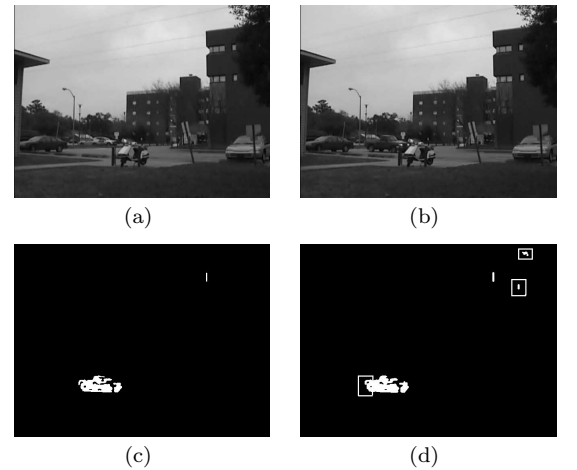


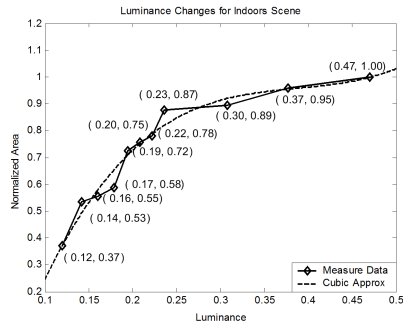
Fig. 10 Object detection results for an outdoor scene a. Background image, b. Image containing a moving object, c. Foreground mask using a 3 x 3 window, d. Foreground mask using a 5 x 5 window.

to perform object detection [21]. 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.

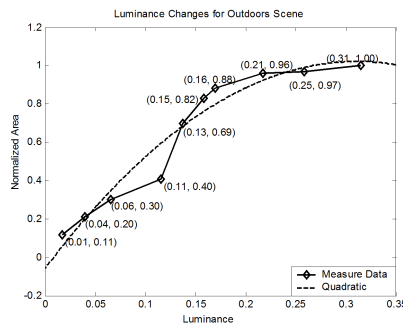
Our simulation uses a frame size of 640 x 480 pixels images. The camera generates 30 frames per second. However it is not really useful to subtract the background for each of these frames. Instead, our simulation took first and last frame each second. This interval reduces timing and power consumption because the operation is performed once each second rather than 30

Table 4 Comparison of approach with previous work.

	Observer Units	Method of Detection	Track Multiple Objects	Tracking Type	Framework	Susceptibility to Network Attack
[7]	Camera are linked to a computer	Adaptive background subtraction and 3 frame differencing	Restricted	Manual	Cooperative Sensors	Susceptible, if OCU fails, the network will fail
[8]	Fixed cameras without processing unit	Classification of moving and non-moving pixels	Yes	Automated	Agent based	Susceptible. Cameras send information to a central unit
Ours	Each AOC contains a camera and processing unit	Wronskian Change Detector	Yes	Automated	ADS and Agent Based	The architecture is an ADS and fault tolerant



(a)



(b)

Fig. 11 Detection performance for indoor and outdoor scene under different luminance values

time.

Background Subtraction algorithm requires vector generation of 9 elements, however the vector dimension could vary. Our simulation considers only windows of 3×3 because the results with larger windows are not found to be extra useful and increases the processing time. Figures 9 and 10 show simulation results, squares enclose the differences between foreground mask resulting with window sizes 3 and 5.

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 11 for indoor and outdoor scenes. The coordinates shown for selected points represent the (luminance, normalized area) pair. Luminance average values give a measure of the global illumination of the scene. Since the system will be deployed on building and parking lots, a medium change of illu-

mination 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. A broad comparison of the proposed approach is done in Table 4 with respect to the existing methods. As seen, the proposed approach is more robust and suited for sensor network because of the ADS framework.

4. Conclusions

A distributed scene understanding architecture has been presented. The architecture consists of several Autonomous Observer Cells distributed on the area under surveillance. Each AOC is attached to a camera. AOCs 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 an Autonomous Analysis Cell, which will classify and decomposed objects. Then, AAC 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 AAC are not loaded of scenes that may not contain useful information.

References

- [1] K. Mori, H. Ihara, Y. Suzuki, K. Kawano, M. Koizumi, M. Orimo, K. Nakai, H. Nakanishi, "Autonomous Decentralized Software Structure and its Application", Proc. of FJCC, pp. 1056-1063, Dallas 1986.
- [2] N.A. Duffie, R. Chitturi, J. Mou, "Fault-Tolerant Hierarchical Control of Heterogeneous Manufacturing System Entities", J. Manufacturing Systems, Vol. 7, No.4, pp. 315-328, 1988.
- [3] A. T. Jones, C.R. McLean, "A Proposed Hierarchical Control Model for Automated Manufacturing Systems" J. of Manufacturing Systems, Vol. 5, No.1, pp. 15-25, 1986.
- [4] I. Ahmed, "Towards the Next Level of Abstraction, The Autonomous Decentralized System- An Overview", Studies in Info. and Control, Vol. 13, No. 4, pp. 295-304, 2004.
- [5] H. Kuwara, "Experience Teach us the Future of Autonomous Decentralized Systems", Proc. Int. Symp. on Auto. Dec. Systems, pp. 169-175, 1997.
- [6] A. R. Dick and M.J. Brooks, "Issues in Automated Visual Surveillance", in Proc. of VIIth Digital Image Comp. Tech. and App., Sydney, pp. 195-204, Dec. 2003.
- [7] R. T. Collins, A.J. Lipton, H. Fujiyoshi, T. Kanade, "Algo-

- rithm for Cooperative Multisensor Surveillance”, Proc. of IEEE, Vol. 89, No. 10, pp. 1456-1477, Oct. 2001.
- [8] P. Remagnino, A.I. Shihab and G. A. Jones. ”Distributed Intelligence for a Multi-Agent Visual Surveillance”, Elsevier Journal of the Pattern Recog. Soc., Vol. 37, No. 4, pp. 675-689, 2004.
- [9] T. Xu-yan, T. Tao, ”Intelligent Autonomous Decentralized System”, Proc. of 2nd Int. Workshop on Auto. Dec. System, pp. 10-15, 2002.
- [10] H. Suguri, E. Kodama, M. Miyazaki, ”Assuring Interoperability in Heterogeneous, Autonomous and Decentralized Multi-agent System”, Proc. of the Sixth Int. Symp. on Auto. Dec. Systems, pp. 17-24, April 2003.
- [11] H. Shibata, T. Hoshia, M. Kubota, M. Teramoto, ”Agent Technology Recommending Personalized Information and its evaluation”, Proc. of 2nd Int. Workshop on Auto. Dec. System, pp. 176-183, 2002.
- [12] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan. ”An Application-Specific Protocol for Wireless Microsensor Networks”, IEEE Trans. on Wireless Comm. Vol. 1, No. 4, pp. 660-670, Oct. 2002.
- [13] Y. Tan, Q. Li, Q. Qian and K. Mori, ”Autonomous Coordination in Sensor Networks: Heterogeneous ADS”, Proc. ISADS’03, China, pp. 208-215, April 2003.
- [14] M. Dietl, J-S Gutmann, and B. Nebel, ”Cooperative Sensing in Dynamic Environments”, Proc. Int’l Conf. Intel. Robots and Systems, pp. 1706-1713, Oct-Nov 2001.
- [15] T. Finin, D. McKay, and R. Fritzson, ”An Overview on KQML: A Knowledge Query and Manipulation Language”, University of Maryland CS Department, Tech. Rep., 1992.
- [16] J. R. Searle, Speech Acts: An Essay in the Philosophy of Language, Cambridge University Press, 1970.
- [17] M.N. Huhns and M.P. Singh, Readings in Agents. San Francisco, Morgan Kaufmann, CA, 1998.
- [18] G. Weiss, Multiagent System: a Modern Approach to Distributed Artificial Intelligence, MIT Press, 1999.
- [19] T. Kamei, ”Face Retrieval by an Adaptive Mahalanobis Distance using a Confidence Factor”, in Proc. Inter. Conf. on Image Proc., Vol. 1, pp. I-153 - I-156, Sept 2002.
- [20] J.C. Collis, D.T. Ndumu, H.S. Nwana, L.C. Lee, ”The ZEUS Agent Building Toolkit”, BT Tech. Journal, Vol. 16, No.3, pp. 60-68, July 1998.
- [21] E. Durucan and T. Ebrahimi, ”Change Detection and Background Extraction by Linear Algebra”, Proc. IEEE, vol. 89, no. 10, pp. 1368-1381, Oct. 2001.
- [22] R. Aguilar-Ponce, A. Kumar, J.L. TecpanecatI-Xihuitl and M. Bayoumi, ”An Architecture for Automated Scene Understanding”, Accepted for Publication, IEEE Inter. Workshop on CAMP Terrasini, Palermo, July 4-6 2005.



Ruth Aguilar-Ponce obtained a Bachelor degree in Computer Science from the Autonomous University of Puebla, Mexico in 1997. She obtained the M.S. in Electronics from the National Institute for Research in Optics, Astrophysics 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 Engineer-

ing in 2004. Currently she is a Ph.D. candidate. Her research interests include sensor networks, communication protocols, system on a chip, signal and image processing.



Ashok Kumar obtained a B.Tech degree in Computer Science and 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 joined the Center for Advanced Computer Studies at University of Louisiana at Lafayette in 1994, where he obtained M.S. and Ph.D. degrees 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 Research and Development operations. Currently he is a Research Faculty at the Center for Advanced Computer Studies at University of Louisiana at Lafayette.



J.Luis TecpanecatI-Xihuitl received the B.E. degree in electrical engineering from the Autonomous University of Puebla, Mexico and the M.S. degree in electronics from National Institute for research on Astrophysics, Optic, and Electronics, Mexico. 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 (CACS) 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 at the Center for Advanced Computer Studies, University of Louisiana at Lafayette, where he has been a faculty member since 1985. 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 was the vice president for the technical activities of the IEEE Circuits and Systems Society. He is the chairman of the Technical Committee on Circuits and Systems for Communication and the TC on Signal Processing Design and Implementation. He won the UL Lafayette 1988 Researcher of the Year award and the 1993 Distinguished Professor award at UL Lafayette. He is a Fellow of the IEEE.