ROBUST AND EFFICIENT EVENT DETECTION FOR THE MONITORING OF AUTOMATED PROCESSES

Thomas Sarmis, Antonis A. Argyros, Manolis I.A. Lourakis, Kostas Hatzopoulos

Foundation for Research and Technology - Hellas (FORTH)
Heraklion, Crete, Greece
{sarmis, argyros, lourakis, hatzop}@ics.forth.gr

Keywords: vision-based event detection, process monitoring

Abstract

We present a new approach for the detection of events in image sequences. Our method relies on a number of logical sensors that can be defined over specific regions of interest in the viewed scene. These sensors measure time varying image properties that can be attributed to primitive events of interest. Thus, the logical sensors can be viewed as a means to transform image data to a set of symbols that can assist event detection and activities interpretation. On top of these elementary sensors, temporal and logical aggregation mechanisms are used to define hierarchies of progressively more complex sensors, able to detect events having more complex semantics. Finally, scenario verification mechanisms are employed to achieve process monitoring, by checking whether events occur according to a predetermined order. The proposed framework has been tested and validated in an application involving monitoring of automated processes. The obtained results demonstrate that the proposed approach, despite its simplicity, provides a promising framework for vision based event detection in the context of such applications.

1 Introduction

With recent advances in computer vision, it is now becoming possible to extract high-level semantic information from video streams. The automatic detection and analysis of events are important in a variety of applications including surveillance, video annotation, vision-based human-computer interaction, etc. For example, the goal of most surveillance systems is the automatic detection of events and suspicious activities for triggering alarms, thus reducing the volume of data presented to a human operator. Road traffic monitoring, airport security, access control to buildings, are just a few out of several important application areas.

Event detection requires the interpretation of the “semantically meaningful object actions” [3]. To achieve this task, the gap between the numerical features of video objects and the symbolic description of their meaningful activities needs to be bridged. Past work has mostly dealt with the extraction of object trajectories followed by supervised learning that makes use of parameterized models for actions [1, 2]. Such models usually consist of predefined dynamic patterns of movements that are learnt in an offline training phase. However, as the nature of events varies depending on the application, event modeling becomes a very challenging task. Currently, there exist several smart camera systems for dealing with several instances of the event detection and activity interpretation problem; some of them are already successful commercial products. Recently, Valera and Velastin [5] provided an extensive review of such camera systems.

In this paper, we propose a new approach to event detection and interpretation. We develop an event detection and process monitoring framework that has two significant advantages over past work. First, it decouples the detection and the interpretation of events from the explicit, computer-based detection and recognition of objects, actions, and their relationships. This is very important because it makes the framework usable in a variety of different application domains. Secondly, the framework depends on very simple, low-level vision processes, which is a key to robust and efficient performance. Despite them working in a different application domain, Ye et al [4] propose a similar approach. Their visual interface cues paradigm relies upon simple image processing to detect pre-defined sequences of visual events and support human-computer interaction.

The proposed approach is based on what we call “Vision Based Logical Sensors” (VBLSs). In the heart of this approach lies the notion of a *Logical Sensor* (LS). A LS is a logical construct that decides (by providing a Boolean, true/false output value) whether a specific property holds in a specific image region at a certain moment in time. Example such properties are "region illumination exceeds predefined threshold", "region changed with respect to the scene background", "region profile matches stored prototype", etc. LSs, enable the detection of primitive events in a video stream. *Compound Logical Sensors* (CLSs) can then be built through *temporal* and *logical aggregation* applied to the output values of LSs (or, recursively, other CLSs). Temporal aggregation creates a CLS by reasoning on the value of a LS (or of a CLS) over time. Logical aggregation creates a CLS by combining the values of several other LSs or CLSs into Boolean expressions. LSs together with the mechanisms of temporal and logical aggregation are used for the detection of events of high complexity.

The framework proposed in this paper is particularly suited to the application area of monitoring of automated processes. In most such processes, things occur in a strict, predetermined way. For example, in an assembly automation process, mechanical parts move on a conveyor belt and are being manipulated by actuators in a process that, typically, presents no considerable deviations. The fact that these processes have considerable structure, permit us to turn difficult detection problems into much simpler verification problems. More precisely, instead of trying to detect what is going on in the viewed scene, the VBLS approach can be used to verify that things proceed as expected. This results in several advantages:

- *Computational efficiency:* The VBLS approach requires simple, low level, computationally cheap, data parallel image processing operations to be applied on (typically) small image regions.
- *Extendibility:* LSs and CLSs can be dynamically tailored and expanded based on the needs of different application domains.
- *Flexibility and adaptability:* Most complex vision algorithms either fail in specific settings or require elaborate, non-intuitive parameter tuning. With the VBLS approach the process of finding an arrangement of LSs/CLSs that succeeds in detecting interesting events, is facilitated.

Moreover, as it will become clearer later in the paper, intuitive explanations at varying levels of detail can be provided by the system while reporting the success or the failure in the detection of certain events.

The rest of the paper is organized as follows. Section 2 provides a description of the basic elements of the VBLS approach. Section 3 describes experiments carried out with a prototype implementation of the approach in the application area of the monitoring of automated processes. The document is concluded with a short discussion summarizing the contributions of this work and with current and future research directions.

2 The proposed VBLS approach

In this section, we present, in more detail, the basic elements of the proposed VBLS approach.

2.1 Logical Sensors (LSs)

A Logical Sensor (LS) is the basic entity in the VBLS approach. Its function is to apply a set of user-defined *Image Processing and Analysis Algorithms* (IPAs) in a user-defined *Region Of Interest* (ROI). The goal of the IPAs is to detect an interesting property within the specific ROI. The output of the LS is a Boolean value, which reports whether the specified ROI has, at a certain moment in time, the property sought by the IPAs associated with this LS.

2.1.1 Region Of Interest (ROI)

A ROI is an arbitrarily shaped, user defined region in an image that is used to spatially constraint image processing and analysis. We denote a ROI *R,* with $R \equiv ROI(I, M, W, H, X, Y)$, meaning a region of interest in image *I*, having a mask image *M* with a bounding box of dimensions *W*x*H,* located at image position (*X, Y*). An image pixel belongs to the ROI if and only if the corresponding mask image pixel has a value of 1. Figure 1 gives an illustrative example of the definition of a ROI.

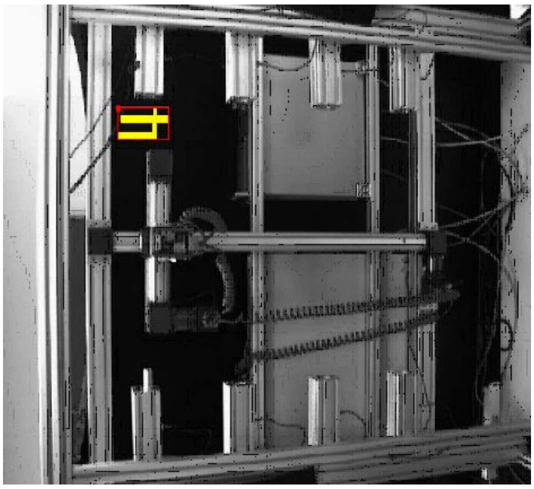


Figure 1: Example ROI (red rectangle). Pixels in yellow color are the ones to be considered for further processing.

2.1.2 Image Processing and Analysis Algorithms (IPAs)

Having defined a ROI, the next step is to define the algorithm(s) that will be applied to it. We differentiate among four categories of IPAs.

- *Preprocessing IPAs:* These algorithms take as input a grayscale image and process it to yield an enhanced/improved image. Examples of such algorithms are various types of filtering operations (Gaussian smoothing, averaging, median filtering, histogram equalization, etc).
- *Analysis IPAs:* They operate on grayscale images and produce a binary image in which pixels that have a certain desired property are differentiated from those that don't have it. Examples of such algorithms are change detection algorithms, template matching, etc.
- *Post-processing IPAs:* These algorithms operate on binary images and produce another binary image with certain desired properties. Examples of such algorithms are the morphological operators (erosion, dilation, etc).
- *Decision IPAs:* These algorithms typically take as input a binary image, and decide the value (true/false) of the Logical Sensor.

A Logical Sensor requires at least one decision IPA to be defined. Preprocessing, analysis and post-processing IPAs are only used to support the decision IPA, if needed. Having mentioned this, a Logical Sensor in the context of the VBLS approach is the (Boolean) result of a collection of IPAs applied over a user defined ROI. More formally, an LS L computes a binary valued function f implemented through a series of IPAs that are applied to a ROI R, or $L \equiv f(R)$.

2.2 Compound Logical Sensors (CLSs)

The Compound Logical Sensors (CLSs) are built based on LSs. There are two mechanisms used to build CLSs. The first mechanism is termed *temporal aggregation*. Time is measured relative to present (*t=0*) and increases towards the past. Moreover, discrete time is assumed, with measurements coinciding with the frame rate of the employed camera. It is assumed that a history of past values is maintained for each of the defined LSs and CLSs. Then, we denote the temporal aggregation of a CLS_j as:

$$CLS_i \equiv TA(CLS_j, A_{\min}, A_{\max}, T_2, T_1) \qquad (1)$$

The meaning of definition (1) above is that a new CLS (CLS_i) is built through temporal aggregation (TA) of the values of a previously defined compound logical sensor, CLS_j. More specifically, CLS_i will be true at time t if CLS_j was true at least A_{min} and at most A_{max} times over the time interval $[t\text{-}T_1, t\text{-}T_2]$. A trivial case of temporal aggregation is of the form $CLS_i = TA(LS_j, 1, 1, 0, 0)$, meaning that CLS_i simply reports the current value of LS_j.

The second mechanism for building CLSs is that of *logical aggregation*. According to this mechanism, a CLS is built based on the logical combination of the results of other LSs (or, recursively, CLSs). The following are some example CLSs:

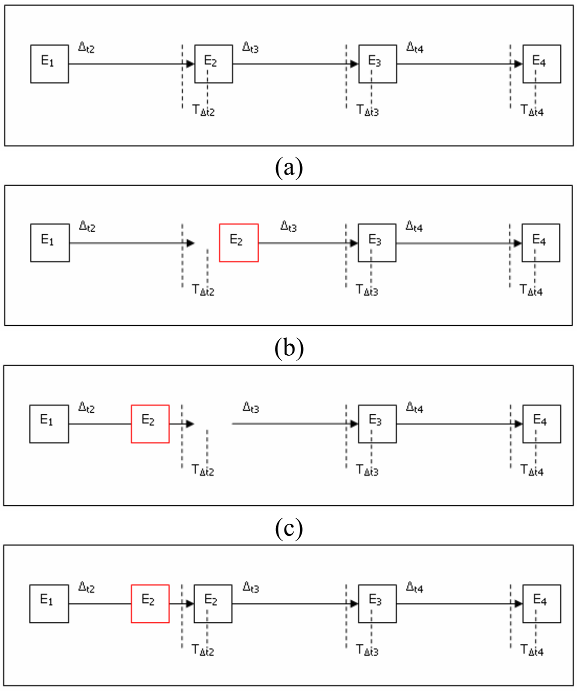
- CLS_1 := LS_1 *OR* LS_2
- CLS_2 := LS_3 *AND* LS_4
- CLS_3 := LS_4 *XOR* LS_5
- CLS_4 := CLS_1 *AND* LS_6 (i.e. CLS_4 is equivalent to the expression "(LS_1 *OR* LS_2) *AND* LS_6").

Logical and temporal aggregation can be combined arbitrarily. Hence, $CLS_7 = TA(CLS_1, 3, 8, 5, 30)$ *OR* $TA(LS_2, 5, \infty, 0, 9)$ is a valid CLS since it is the result of the logical aggregation of two CLSs, each resulting from the temporal aggregation of other CLSs and LSs. Moreover, CLS_7 will be true if CLS_1 is true for at least 3 and at most 8 time instances in the time interval $[t-30, \ \ t-5]$ or if CLS_2 is true at least 5 times during the last 10 time instances.

2.3 Scenarios

CLSs, as described in the previous section, can be used to combine several measurements in (image) space and time to detect interesting events. The *scenarios* are mechanisms provided to support the automatic monitoring of processes that consist of several events occurring serially, one after the other. A scenario SC is defined by the ordered list E of events $E_1, E_2, \ldots, E_n$ comprising it, the time differences d_i between the successive events E_i and E_{i+1}, and the time tolerances τ_i in the occurrence of these events. This means that if the event E_i occurs at time t_i, then, according to the scenario, the event E_{i+1} should occur in the time interval $[t_i + d_i - \tau_i, \ \ t_i + d_i + \tau_i]$. More formally, a scenario SC is represented by the triplet $SC \equiv (E, D, T)$, where $E = \langle E_1, E_2, \ldots, E_n \rangle$, $D = \langle d_1, d_2, \ldots, d_{n-1} \rangle$ and $T = \langle \tau_1, \tau_2, \ldots, \tau_{n-1} \rangle$. The validation of a scenario is achieved by a mechanism that checks whether the events comprising the scenario occurred with a timing that respects the rules set. There are two different types of scenarios depending on the scheme used for their validation. In the case of a *strict scenario,* the events comprising it should occur only with the predetermined timing. In the case of a *relaxed scenario*, the events should occur *at least* with the predetermined timing; however, some of the events could also occur at other time instances, besides the ones specified in the scenario.

Figure 2 shows three scenario examples. Figure 2(a) shows the scenario as defined by the user. The vertical lines correspond to the time intervals in which each event (E_i) must occur. In Fig. 2(b) event E_2 occurs after the defined interval. Both relaxed and strict scenario verification mechanisms will detect the failure after the end of the interval. In Fig. 2(c) E_2 occurs before the predefined interval. The strict scenario verification mechanism will detect the failure the moment E_2 occurs. The relaxed mechanism will detect the failure at the end of the interval. Finally, in Fig. 2(d) the event E_2 occurs twice, both before and within the interval. In this case the strict mechanism will report a failure the moment E_2 occurs for the first time. The relaxed mechanism does not consider the premature occurrence as a fault therefore this scenario is considered as a successful one.



(a)

(b)

(c)

(d)

Figure 2: Schematic illustration of successful and unsuccessful relaxed and strict scenarios.

2.4 Error reporting mechanism

Regardless of its type, a scenario may fail either because an event was never detected, or because it was detected but did not occur with the proper timing. In both cases the framework may provide an intuitive explanation for scenario failure, at different levels of detail. This is achieved by tracing the hierarchical structure of the CLS responsible for the non-detected event and reporting the lower-level CLS or LS that did not produce the expected value.

Figure 3 displays the operation of the error reporting mechanism. When an error is detected, the error reporting mechanism is used to provide information regarding the failed scenario, the event that caused the failure, and the compound logical sensor that corresponds to that event. Also the report contains all the CLSs and LSs that may have affected the result of the CLS that corresponds to the detection of the failed event.

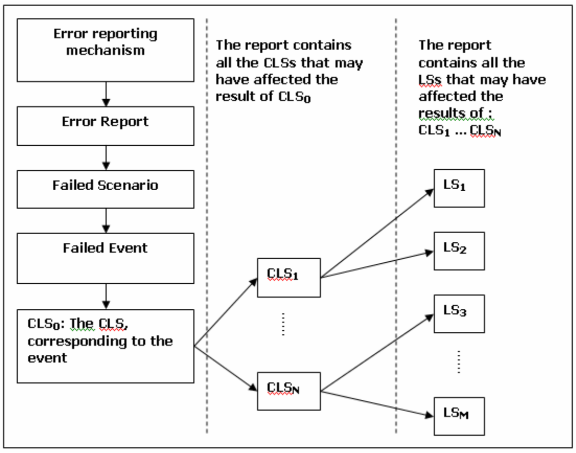

Error reporting mechanism
Error Report
Failed Scenario
Failed Event
CLS0: The CLS, corresponding to the event
The report contains all the CLSs that may have affected the result of CLS0
CLS1
CLSN
The report contains all the LSs that may have affected the results of : CLS1 ... CLSN
LS1
LS2
LS3
LSM

Figure 3: Error reporting mechanism

2.5 Overview of the proposed approach

Figure 4 shows a graphical presentation of the overall VBLS approach. The logical sensors process the image and create primitive information. The compound logical sensors can detect high level events by combining the primitive results created by LSs, using temporal and logical aggregation. Scenarios are constructed based on those events and scenario verification mechanisms are used to verify that events occur according to the predetermined timing. When a failure is detected, the error reporting mechanism is used to produce an error report.

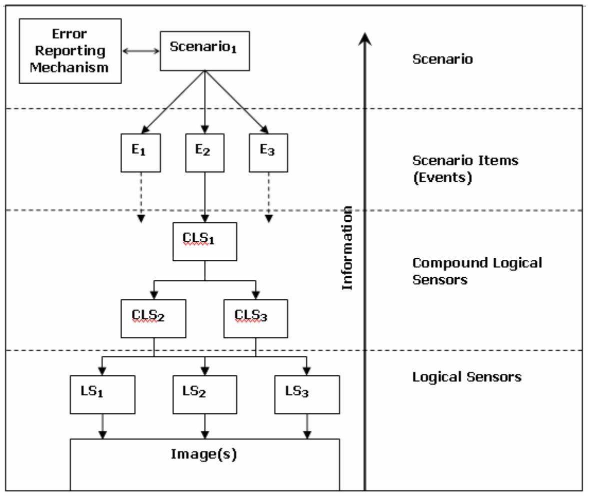
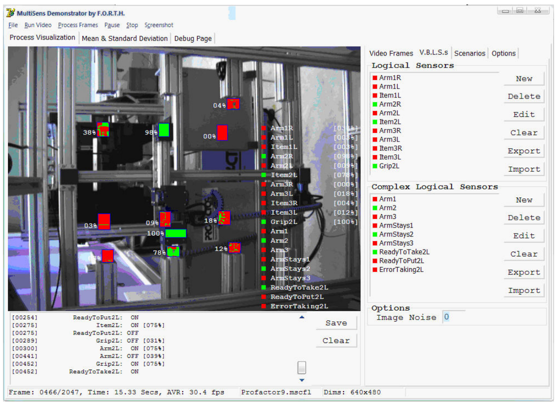


Figure 4: Overview of the VBLS approach

3 Experiments

A software platform has been developed in order to test and validate the VLBS approach. The capabilities of the platform include image sequence visualization, definition of ROIs, definition of IPAs, definition of LSs, definition of CLSs, definition of scenarios (both strict and relaxed), parameter tuning, control of several visualization options, saving of results in textual/video form and detailed error reporting.



```
[00254]         ReadyToPut2L:  ON
[00275]              Item2L:  ON [075%]
[00275]         ReadyToPut2L: OFF
[00289]              Grip2L: OFF [031%]
[00300]               Arm2L:  ON [075%]
[00441]               Arm2L: OFF [039%]
[00452]              Grip2L:  ON [075%]
[00452]        ReadyToTake2L:  ON
```

```
Frame: 0466/2047, Time: 15.33 Secs, AVR: 30.4 fps    Profactor9.mscfl    Dims: 640x480
```

Figure 5: The software developed for testing the VBLS approach, while in operation.

The VBLS approach has been tested in the context of an application involving the monitoring of the activities of a 5-axis robot. The task of this robot is to move parts (e.g. a drill) between different tool fittings using a gripper. A camera system was set up to observe and monitor the operation of the robot. Using the VBLS software platform, a user with limited training was able to quickly define LSs, CLSs and scenarios to detect and verify several interesting modes of robot operation (complex object manipulation, part relocation, etc). Moreover, the system was able to detect several failure situations, such as the dropping of the drill during relocation, failure of the gripper to remove the drill from a particular position, etc. In several of the conducted experiments a human has been entering the working space of the robot and the camera field of view. Nonetheless, this did not affect the correctness of the system's operation. Figure 5 shows a typical screenshot of the system while in operation. LSs and CLSs which happen to be true or false at the particular moment in time are shown in green or red color, respectively.

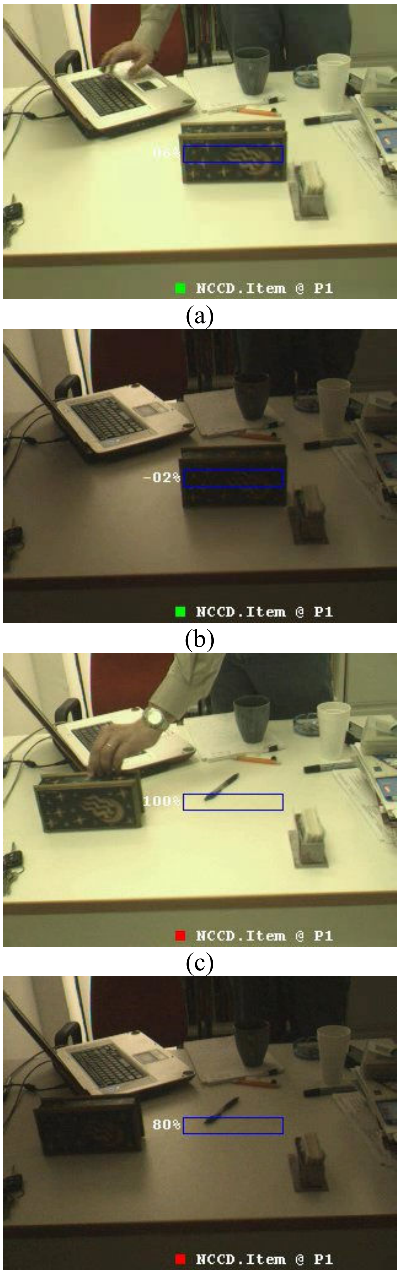
Another experiment has been conducted to assess the performance of the proposed approach under varying illumination conditions. A video has been recorded in which a person manually changed the location of an object on a table. While doing this, the lighting conditions were changing significantly by turning the room lights on and off. A logical sensor has been defined performing change detection through Normalized Cross Correlation. Figure 6 shows characteristic snapshots from this experiment. As it can easily be verified, this logical sensor is able to detect the presence/absence of the object despite the significant illumination variations. This demonstrates that by incorporating simple, low level vision algorithms into the VBLS approach considerable invariance to illumination changes can be achieved.

Several other experiments have been conducted involving image sequences acquired from real-world situations in assembly automation. Preliminary results are very promising.

It is interesting that the current implementation of the VBLS framework involves only very simple IPAs such as Gaussian smoothing, median filtering, Gaussian background modeling, and various thresholding operations. This means that the power of the framework in detecting complex events lies in the spatial and temporal aggregation of the information of a large number of logical sensors rather than in the "intelligence" of one, complex vision module. We consider this to be an important property of the developed framework that can lead to efficient and robust performance in a wide variety of application domains.

4. Summary

In this paper, we have proposed a new approach for event detection and process monitoring. The aim of the



(a)

(b)

(c)

(d)

Figure 6: (a), (b) the logical sensor detects the presence of the object in its ROI under different illumination conditions, (c), (d) the same logical sensor detects the absence of the object under varying illumination conditions.

approach is to enable the spatio-temporal integration of measurements taken from critical areas in the viewed scene in order to be able to detect events and to monitor processes. The proposed approach has been tested in the context of an application involving monitoring of automated processes. The proposed framework seems ideally suited for this application domain because events and activities therein have considerable structure. The preliminary results are very promising, as they reveal a number of beneficial framework properties, including but not limited to computational efficiency, extendibility, flexibility and adaptability. An interesting extension is to become able not only to monitor such processes but also to control them. This can be achieved by substituting the signals driving a robot or machine by the signals provided by the various LSs and CLSs. This would replace a number of hardware, expensive and difficult to reconfigure sensors with a virtual, cheap and easy-to-reconfigure sensor network defined in the field of view of a single camera. Preliminary experiments in this direction gave very promising results. It also remains to be verified whether the same framework can be useful in less structured scenarios such as those involved in traffic monitoring, human-computer interaction and other surveillance applications.

Acknowledgements

This work was partially supported by the EU Project MultiSens. The authors want to thank Profactor GmbH, for providing the hardware testbed and image sequences used to test the VBLS approach.

References

[1] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, R. Nevatia, “Event Detection and Analysis from Video Streams”, IEEE Trans. on PAMI, vol. 23 no. 8, Aug. 2001, pp. 873-889.

[2] S.O. Orphanoudakis, A.A. Argyros, M. Vincze “Towards a Cognitive Vision Methodology: Understanding and Interpreting Activities of Experts”, ERCIM News, No 53, Special Issue on “Cognitive Systems”, April 2003.

[3] Fatih Porikli, “Trajectory Distance Metric using Hidden Markov Model based Representation”, PETS-ECCV’04 Workshop, Prague, Czech Republic, May 2004.

[4] G. Ye, J. J. Corso, D. Burschka and G.D. Hager, “VICs: A Modular HCI Framework Using Spatio-Temporal Dynamics”, **Machine Vision and Applications**, 16(1):13-20, 2004.

[5] M. Valera and S.A. Velastin, “Intelligent Distributed Surveillance Systems: A Review”, IEE Proc.-Vis. Image Signal Process., Vol. 152, No. 2, April 2005, p.192-204.