On the Effect of Feedback in Multilevel Representation Spaces for Visual Surveillance Tasks

Enrique J. Carmona[1], Mariano Rincón[1], Margarita Bachiller[1], Javier Martínez-Cantos[2], Rafael Martínez-Tomás[1], José Mira[1]

[1]Dpto. Inteligencia Artificial. E.T.S.de Ingeniería Informática,
Universidad Nacional de Educación a Distancia,
Juan del Rosal 16, 28040 Madrid, Spain
{ecarmona, mrincon, marga, rmtomas, jmira}@dia.uned.es,

[2]Dpto. de Sistemas Informáticos, Escuela Politécnica Superior de
Albacete & Instituto de Investigación en Informática de Albacete (I3A),
Universidad de Castilla-La Mancha, 02071 – Albacete, Spain
javier.mcantos@uclm.es

Abstract

In this work we propose a general top-down feedback scheme between adjacent description levels to interpret video sequences. This scheme distinguishes two types of feedback: repair-oriented feedback and focus-oriented feedback. With the first it is possible to improve the system's performance and produce more reliable and consistent information, and with the second it is possible to adjust the computational load to match the aims. Finally, the general feedback scheme is used in different examples for a visual surveillance application which improved the final result of each description level by using the information in the higher adjacent level.

Key words: Visual Surveillance Systems, Video Sequence Interpretation, High Level Vision, Predictive Diagnosis Task, Feedback between Description Levels.

1. Introduction: surveillance based on High Level Vision

High Level Vision (HLV) is defined as scene interpretation rather than just mere object recognition and tracking [7]. This implies the need to recognise situations, activities and interactions among the different agents participating in a scene. It is a question of linking the physical signals that reach a camera's sensors with the interpretation of their meaning. When a human observer interprets the meaning of a scene, obviously, he uses his knowledge of the world, the behaviour of the things that he knows, the laws of physics and the set of intentions governing the agents' activities. All this additional knowledge that does not appear explicitly in the video camera signals enables the observer to model the scene and use this model to predict, at least partially, what may happen. In other words, to cast hypotheses about the temporal evolution of the events and activities detected. It therefore seems logical that we also have to use an explicit and declarative representation of this additional knowledge not included in the signals to design an HLV system. In particular, the dynamic models of objects of interest and different behaviour patterns constitute the events and activities of interest for the task into which the process for understanding images is going to be integrated.

To represent this knowledge, the set of techniques available in Artificial Intelligence (AI) are used: logic, rules, graphs, finite automata, frames, agents, Bayesian networks, neural networks, task structural models, task breakdown methods and a set of static and dynamic roles with which it is possible to complete the generic model with specific knowledge of each application domain [6][9][13][21][23].

The visual surveillance task affects an increasing number of scenarios, services and clients. It implies observing areas considered at risk or vulnerable, where thefts, vandalism, bomb attacks, or any other dangerous event may occur. The spectrum of situations and needs is very wide: from the mere detection of movement in a controlled space to an integral control system of the scene from multisensory signals, which would include: (1) diagnosis of the situation displayed and control on actuators searching for new data and findings, and (2) dynamic planning (according to new diagnoses resulting from partial actions) of the actions consistent with the resources available [11]. When an HLV system is integrated into a surveillance task, information of different representation levels travel bottom up (from the pixels to the activities) and top down (from the activities to the blobs). This top-down feedback is used to improve the lower-level tasks, such as segmentation, the identification of objects of interest and the tracking of their trajectories. In this paper, we will focus on this feedback.

The following works refer to using representation spaces in bottom-up organisation, equivalent to passing from the retinal photoreceptors to retinotopic projection in the primary cortex. Thus, Neumann [16][18] described a system for generating a natural language description of the activities observed in a traffic video sequence, using frames of examples based on locomotion verbs organised hierarchically for the representation. Buxton and Gong [2] address the need to deal with uncertainty and use contextual information to improve the detection and tracking of vehicles and people. Bobick [1] characterises movement in terms of consistency of the entities and relations detected in a time sequence. In contrast, the concept of activity is understood as "a statistical sequence of movements", i.e., a composition of stereotyped movements, whose time sequence is characterised by statistical properties (e.g. hand gesture). Finally, he defines action as "semantic primitives relating to the context of the motion". In [15], a hierarchical ontology is structured by Nagel (*events*, *verbs*, *episodes*, *stories*, etc.). On the other hand, Chleq and Thonnat [4] only distinguish between primitive and composed events. All these works use context and the injection of complementary knowledge to link the physical signals with the underlying actions. Generally, the activities or more abstract events are thus considered to be a composition from other more primitive lower-level events. This composition is done with spatial-temporal relations, like in [19], or with common-sense knowledge on hierarchies and concept relations, like in [13].

However, in neuroscience the complementary effect of bottom-up organisation of perception is well known, which includes all the processes of selective visual attention and specification of the characteristics defining the events and objects of interest. Computationally, this implies considering different feedback loops from the highest to the lowest semantic levels to improve the specific tasks of these levels. There are specific references to using top-down feedback in high-level vision in the works of Howarth&Buxton [8], Rincón et al [20] and Rota&Thonnat [21].
In this paper, we illustrate the positive effect of feedback for three specific examples. (1) Use of object-level knowledge to improve segmentation in the blob level, thereby

reducing the noise underlying the initial segmentation (bottom-up). This makes it possible to improve the detection of areas as a result of the emergence of new object pixels that before feedback would not have been considered as belonging to this object. (2) Feedback from the activity level to the object level to improve the detection of events which, previously, in bottom-up organisation, had not been considered as such from the object level information. In particular, we shall see how feedback resolves inconsistency problems. (3) Feedback from the activity level to the object level with the aim of focusing, centring computational resources on the search for new findings necessary to fully understand the activities.

In this paper we do not aim to present an evaluated prototype in a wide repertoire of scenarios but to illustrate specifically, in our prototype, the effectiveness of top-down feedback in a multilevel representation space. Thus, we shall stress the basic method for explicitly resolving prototype tasks selected in the level, and the effectiveness of feedback for improving this task. While in earlier works we introduced our approach to bottom-up organisation [3][10][12], in this work we explore the usefulness of top-down feedback that uses the specific knowledge in each level to improve the tasks in the lower levels.

The rest of the work is structured as follows. Section 2 explains the structure of the description levels proposed for interpreting video sequences and how this structure is integrated into the surveillance task. Also a general top-down feedback scheme between adjacent description levels is proposed. Section 3 shows examples characteristic of the different types of description-level feedback, where the system's effectiveness is substantially improved. The interface for monitoring and tracking of the scene is shown, and each of the four levels is described in its own language. Section 4 summarises the conclusions.

2. Communication between Description Levels

There is agreement in the area, within a varied and dispersed nomenclature, on facilitating the large semantic gap between the physical signal and the knowledge level by breaking it down into several intermediate description levels with an increasing degree of semantics [10][14][17]. This enables knowledge to be injected at the right level and environment information (physical, behavioural or social environment, knowledge of the task, etc.) to be projected on each of these intermediate levels. In particular, following the proposal of [14], in our works we distinguish between pixel, blob, object and activity levels (Fig. 1). Each of these levels is modular and independent, and the information handled comes from the ontology of the own level and from the adjacent levels. In the blob level, the entities are associated with the visible part of objects of interest, the blobs. In the object level the information associated with blobs for producing a description of the objects of interest on the scene is reorganised. The models of the objects of interest are described here, which contain: 1) the visual characterisation of the object and its spatial-temporal evolution; 2) the composition relations used to describe complex objects; and 3) the relations between objects for generating the geometric task-oriented description of the scene. The activity level constructs complex events (activities) using the set of predefined primitive events from the object level. These primitive events emerge from measurements on the morphology and trajectory of the objects identified by visual operators (identification and tracking).

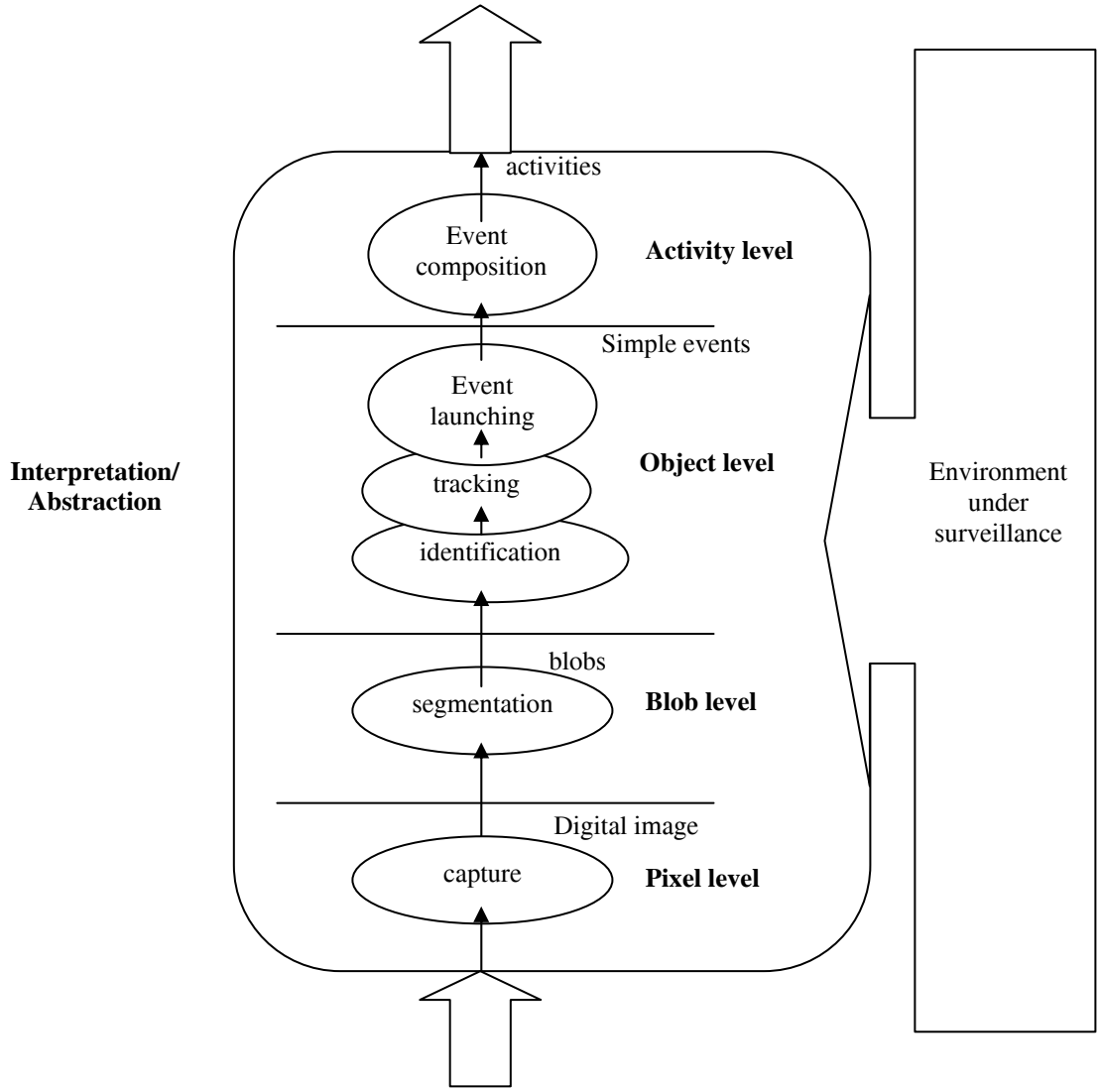

Fig. 1 Description levels and processes in each level, from the physical level to cognitive interpretation of the scene.

In addition to the emerging interlevel communication, which corresponds to a flow of bottom-up information from each level (*i*-1) to its immediately higher level (*i*), in this work we explore the complementary function of top-down organisation in the creation and updating process of a multilevel representation space. In other words, an information feedback scheme that descends from each level (*i*) to its immediately lower level (*i*-1), as shown in Fig. 2.

The emergence process arises as a direct consequence of the breakdown of the task into different levels of abstraction and, therefore, the need to transmit the new information that is generated at each level to the immediately higher level. If the output information in each level were complete and error free, the information flow between levels would be solely and exclusively emergent. However, these two hypotheses are

not fulfilled in real application domains. In fact, the output in each level is subject to the presence of noise, which may falsify the processing in the next level, or the information provided by a level may be incomplete, which would prevent processing in the next level because of insufficient data. Consequently, it is necessary to introduce two new interlevel communication mechanisms with which it is possible to tackle the effects of the two aforementioned problems. Naturally, we are speaking of two feedback processes: *repair-oriented feedback* and *focus-oriented feedback*. The first will eliminate errors and the second will refine the already existing information.

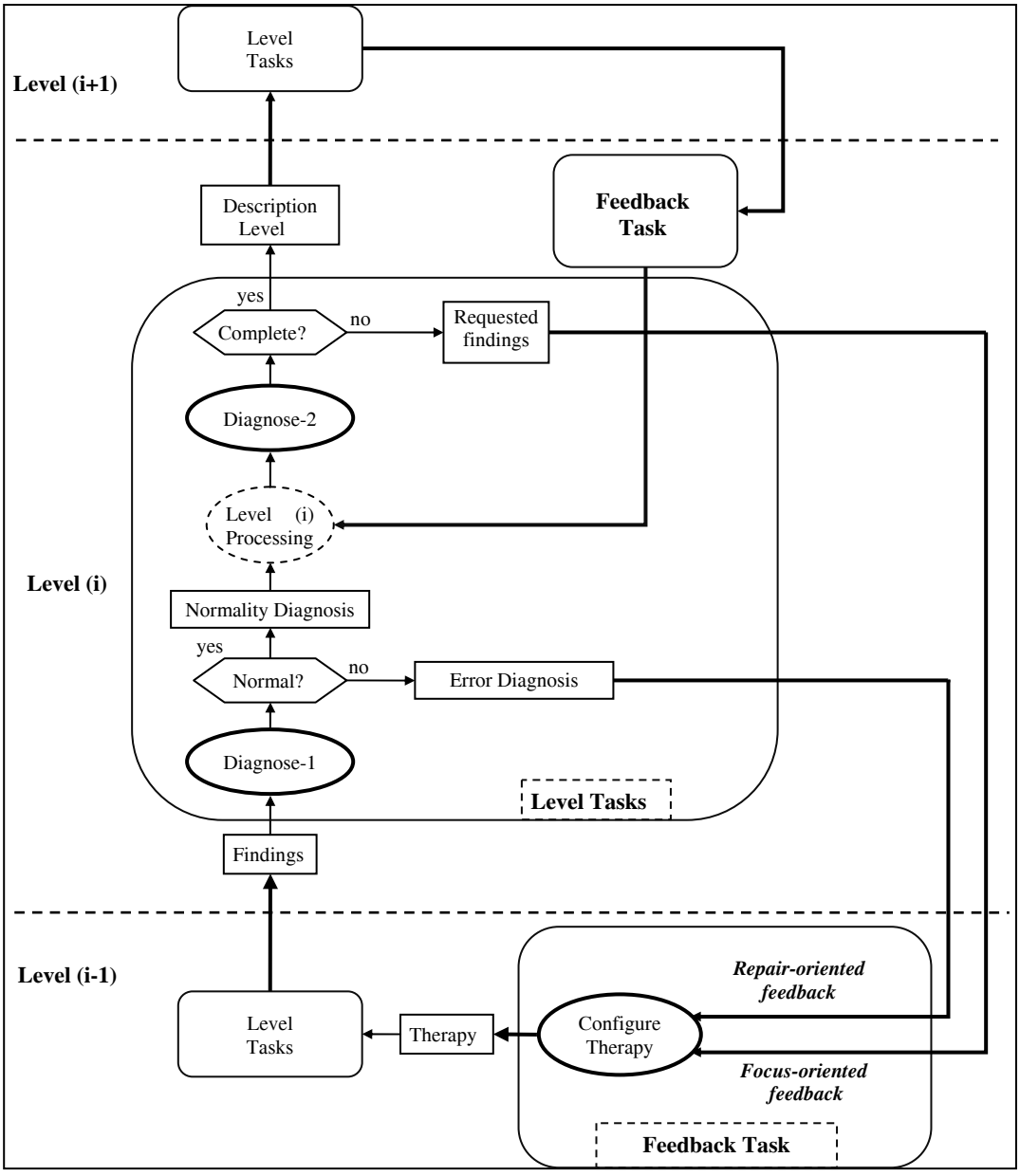

Fig. 2. Generic scheme showing the emergent and feedback information flow between adjacent levels.

To model these two feedback mechanisms, we have included two *diagnosis* tasks. In the first (*Diagnose-1*), from the output information from level (i-1), which constitutes the input (*Finding*) to level (i), a diagnosis is done to evaluate the consistency of the new information with the knowledge of the level. In case of error (*Error Diagnosis*), it is necessary to go back to level (i-1) to select the appropriate therapy (*Configure Therapy*) and repair the error detected. In the second diagnosis (*Diagnose-2*), the completeness of the results obtained from the *level processing* is assessed and, if the information is incomplete (*Requested-Findings*), it is necessary to go back to the previous level to complete this information and select new operators (*Therapy*).

3. Examples of surveillance system feedback

This section describes examples of feedback between the different adjacent levels used in our surveillance system and how this feedback solves different problems and substantially improves the system's performance.

3.1. Repair-oriented Feedback between the Blob and Object Levels

The feedback that we propose in this section, between the blob and object levels, aims to improve the segmentation task output belonging to the blob level with information provided by the object level. Fig. 3 shows the different stages in this process and the information flow. This figure is obtained by instantiating the generic scheme shown in Fig. 2. Thus, the segmentation process begins by taking a frame of the video sequence as input. The result of this initial segmentation will be a first segmentation proposal consisting of obtaining a foreground map of the set of blobs associated with moving objects on the scene. The quality of this set of objects is assessed using the diagnosis task (*Diagnose-1*), which takes a normality model as a reference, distinguishing between normal situations (*Normality Diagnosis*) and anomalies (*Error Diagnosis*). A normality diagnosis means that there are no anomalies in the blob level segmentation and, consequently, all the object level processing can be done with the segmentation result as input. In contrast, if an anomalous situation is detected, a therapy is proposed to act on the segmentation process. In the new context proposed by the therapy, the segmentation stage is executed again, thereby producing an output where the errors detected are repaired. The feedback cycle is accordingly completed. Each of the stages of this process is described in detail below.

Initial segmentation

In any segmentation method of moving objects based on background subtraction, the main demonstrations of noise are associated with foreground noise (shadows, reflections, ghosts, fluctuation) and background noise (moving object pixels that are not detected). In our surveillance system, we use the *truncated cone method* (TCM) [3] as the segmentation method. Given that with this method it is possible to eliminate a large part of the foreground noise, here we only focus on describing how to eliminate the background noise from the scene using the repair-oriented feedback mechanism.

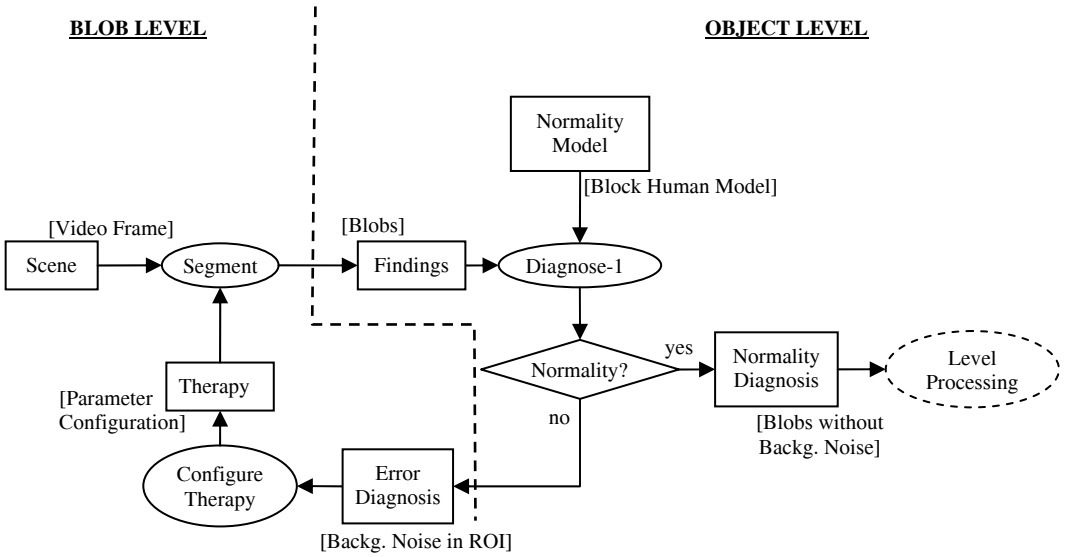

Fig. 3. Repair-oriented feedback structure for segmentation improvement.

The basic idea of the TCM is to transform the three-dimensional RGB colour space into a new two-dimensional space called angle-module space. The new resulting space allows us to define a segmentation rule, represented by Eq. (1), whose application produces a foreground map denoted by $\mathbf{F_t}(x,y)$. In this equation, t corresponds to the instant of current time; $\mathbf{\Theta_t}(x,y)$ is the angle matrix, in which each element represents the angle forming the image RGB vector, $\mathbf{I_t}$, and the background model RGB vector, $\mathbf{B_t}$; $\mathbf{\Delta^t_{mod}}(x,y)$ is the matrix formed by the difference in the modules $\mathbf{I_t}(x,y)$ and $\mathbf{B_t}(x,y)$, in absolute value and, finally, the constants ω_0 and h_0 are threshold values.

$$F_t(x,y) = \begin{cases} 1, \text{if } \Theta_t(x,y) > \omega_0 \text{ or } \Delta^t_{mod}(x,y) > h_0 \\ 0, \text{otherwise} \end{cases} \quad (1)$$

Geometrically, the underlying idea in Eq. (1) is the following: for each point of the image, (x,y), a revolution cone can be built in the RGB space (see Fig. 4.a) by using the straight line containing $\mathbf{B_t}$ as an axis, and another straight line as a generatrix which, passing through the origin, forms an angle ω_0 with the other straight line. If we now trace three planes perpendicular to vector $\mathbf{B_t}$, one containing the point $(r,g,b)_B$, and the other two, situated above and below this, at a distance h_0, they will delimit, along with the cone surface, two regions of interest: a truncated cone located on the upper part of the intermediate plane and another on the lower part. It is quite immediate that if we choose sufficiently small ω_0 and h_0, it is possible to establish as a movement condition that every point of the image, $(r,g,b)_I$, which is outside the truncated cone region, will belong to a real moving object. Most of the remaining points inside this region correspond to fluctuation noise.

The use of the angle-module space facilitates the characterisation of foreground noise and, consequently, its elimination [3]. However, the truncated cone region mentioned earlier not only contains points belonging to fluctuation noise, but also includes all the moving object pixels whose intensity is very similar to their corresponding background model pixels and they are not detected (background noise). Therefore, the problem

posed is how to fine-tune the value of the parameters, ω_0 and h_0, in order to separate the two types of pixels mentioned. The strategy proposed here is to include new parameters that make it possible to delimit new subregions in the original truncated cone volume. For example, the inclusion of the new thresholds h_{01}, h_{02} and ω_1, as defined in Fig. 4.b, make it possible to delimit, along with h_0 and ω_0, new subregions. The right choice of threshold values will be sufficient to separate the pixels belonging to background noise from the ones belonging to fluctuation noise.

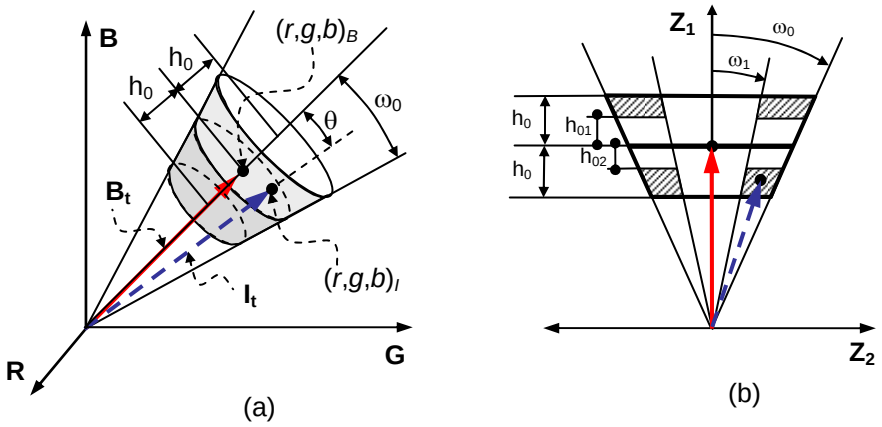

Fig. 4. Truncated cones associated with a point of the background model in (a) the RGB space and (b) as a projection on the Z_1-Z_2 plane (the Z_2 axis is constructed to coincide with the background RGB vector). $\boldsymbol{B}_t$ and $\boldsymbol{I}_t$ represent the background RGB vector and the image RGB vector, respectively.

As will be seen in the next two subsections, the idea is to use the diagnosis task to determine those regions of the scene which have a high probability of containing background noise. Immediately after, the therapy configuration task will fine-tune the threshold values.

Diagnosis

As shown in Fig. 3, the set of blobs (findings) resulting from the segmentation process are used as input in the object level and taking a *normality model* of the kind of moving objects of interest as a reference, the diagnosis task assesses whether the result obtained in the segmentation is consistent with this model.

In this work, we have restricted the type of objects of interest to humans. Consequently, an approximate model of the human is required to use it as a reference model. The human model used here [5] is based on a *block model*. Basically, it consists of dividing a human's blob vertically into six regions of the same height (Fig. 5). Each of these regions is defined by the rectangle around it called *block*. Conceptually, the blocks in this division correspond to areas related to the physical position of specific parts of the body (head, hands, feet, trunk) when a human performs habitual actions. The main advantage of this division is that it enables us to study the human in parts. Thus, with this model it is possible to detect distinct anomalies in the segmentation of a human silhouette, like the disappearance of some significant part that may cause fragmentation of the silhouette into several unconnected blobs, unjustified changes in the segmentation from a frame to the next or the presence/absence of blobs not related to the human (foreground/background noise).

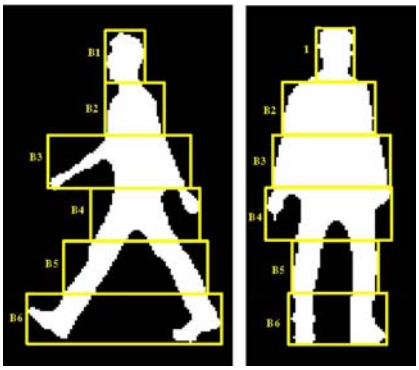

Fig. 5. A human's blob in frontal and lateral position divided into blocks.

Using this human model as a reference, the diagnosis task will search to see whether any block exists associated with this model with a significant number of pixels missing (error diagnosis). If it does, this region of the image will be proposed as the region of interest (ROI) where the search for the missing pixels must be done. Note that, although the process was described just using one block of the human model with background noise, really there is no limit whatsoever to the number of boxes that can be re-fed to the lower level.

Therapy configuration

To fine-tune the thresholds used by the segmentation method in the ROI determined in the diagnosis stage, each of the *angle-module* space dimensions will be analysed independently. For each dimension, the input ROI histogram will be compared with a background model histogram. Thus, an increased number of pixels in some regions of the first histogram will correspond to pixels associated with moving object blobs.

We are going to focus on analysing the *angle* dimension, because the analysis for the module is similar. In the first place, it is defined a mask M^{B}_{ROIi} (Eq. (2)) that allows us to select those pixels in the input ROI, ROI_i, that have initially been classified as background.

$$M^{B}_{ROIi} = \{(x,y) \in ROI_i \mid \mathrm{F_t}(x,y) = 0\} \tag{2}$$

We shall use this mask to select the elements, V^{θ}_{Mi}, from the angle matrix, $\Theta_t(x,y)$, that will form part of the analysis (see Eq. (3)).

$$V^{\theta}_{Mi} = \{\theta_{ij} \in \Theta_t \mid (i,j) \in M^{B}_{ROIi}\} \tag{3}$$

The normalised histogram of V^{θ}_{Mi} with regard to angle, NH^{θ}_{Mi}, is calculated according to Eq. (4), where *hist(list,range,N)* represents the histogram function, which groups the list of values, *list,* in N bins of the same size into which the space of possible values (*range)* is divided, and the function *card(x)* returns the cardinal of set *x.* In our example, *range* = $[0,w_0]$, where w_0 is the value obtained from Eq. (1) and $N = 100$.

$$NH_{Mi}^{\theta} = \frac{hist(V_{Mi}^{\theta},[0,\omega_0],\mathrm{N})}{card(V_{Mi}^{\theta})} \tag{4}$$

Furthermore, to calculate the background model histogram an extended ROI is taken, ROI_e, resulting from expanding the ROI_i a certain offset, δ, in height and width. From ROI_e and following the same steps described in Eqs. (2), (3) and (4), we obtain the mask, M_{ROIe}^{B}, the reference elements, V_{Me}^{θ}, and the normalised histogram, NH_{Me}^{θ}. Figure 6 shows an example of normalised histograms obtained for ROI_i and ROI_e for the parameter *angle*.

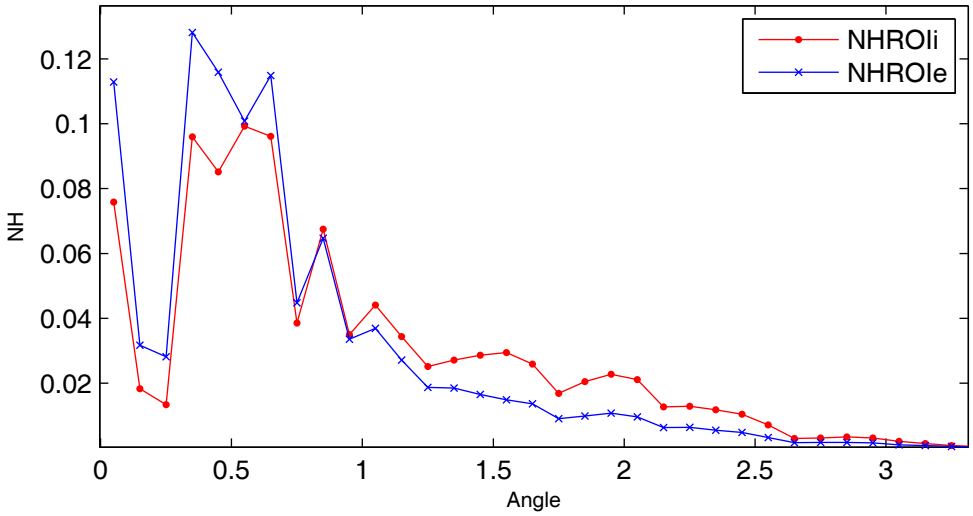

Fig. 6. Normalised histograms corresponding to the parameter *angle* of the input ROI, NH_{Mi}^{θ}, and the extended ROI, NH_{Me}^{θ}.

The next step is to analyse both histograms in order to classify the ROI_i pixels in background and foreground pixels. If we assume that the effect of foreground pixels is negligible in NH_{Me}^{θ}, for each bin j, Eq. (5) defines the relation of background and foreground pixels in this bin with the normalised histograms corresponding to ROI_i and ROI_e, where p_F^j are the pixels in bin j belonging to the object, p_B^j are the pixels that are still background and p_{BF}^j are pixels belonging to the background according to the background distribution obtained from ROI_e with NH_{Me}^{θ} but that are in the area occupied by the foreground object.

$$\frac{NH_{Mi}^{\theta}(j)}{NH_{Me}^{\theta}(j)} = \frac{p_F^j + p_B^j - p_{BF}^j}{p_B^j} \tag{5}$$

Those pixels belonging to the object will increase the normalised histogram in specific bins, therefore, if the bins j where $NH_{Mi}^{\theta}(j) > NH_{Me}^{\theta}(j)$ were selected, a segmentation would be obtained that would contain the object pixels, but also pixels corresponding to background in the selected bins. To reduce this undesired effect, the following approximation is used. The bins j are ordered in decreasing order according to

the ratio between $NH^{\theta}_{Mi}(j)$ and $NH^{\theta}_{Me}(j)$ and the first B bins are selected, until the constraint expressed by Eq. (6) ceases to be fulfilled. This constraint establishes a relationship between the foreground and background pixels included in the new segmentation. A K value close to 1 will ensure that all the object pixels are detected, but it will also segment much background. Currently, the K value is pre-selected with a commitment value in the range [1.5, 3].

$$\frac{\sum_{j=1}^{B} \frac{NH^{\theta}_{Mi}(j)}{NH^{\theta}_{Me}(j)}}{B} > K \tag{6}$$

Finally, to reduce the computational cost, only those bins are selected that form part of groups of more than three consecutive bins, thereby obtaining n groups of bins. For each group of bins, the ends define a range $[w_a,w_b]$ of the parameter *angle* with which a partial segmentation, F_i^{ω} ,is obtained acting on the parameter *angle* according to Eq. (7), where $i = 1,...,n$.

$$\mathbf{F}_i^{\omega}(x',y')\Big|_{wa}^{wb} = \begin{cases} 1 & \text{if } \omega_a > \boldsymbol{\Theta}_{\mathbf{t}}(x',y') > \omega_b \text{ and} \\ & (x',y') \in M^{B}_{ROIi} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The same analysis is done for the parameter *module*. In this instance, the ranges will be defined in accordance with the end bins $[h_a,h_b]$ and the partial segmentation will obey Eq. (8), where $j = 1,...,m$.

$$\mathbf{F}_j^{h}(x',y')\Big|_{h_a}^{h_b} = \begin{cases} 1 & \text{if } h_a < \boldsymbol{\Delta}^{\mathbf{t}}_{\mathbf{mod}}(x',y') < h_b \text{ and} \\ & (x',y') \in M^{B}_{ROIi} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The final result of the segmentation will be achieved by joining the initial segmentation and all the partial results obtained from applying Eqs. (7) and (8) to the selected ranges of each parameter, as indicated in Eq. (9), where (x,y) represent the coordinates of any point of the frame and (x',y') within the ROI_i.

$$\mathrm{F}_{feedback}(x,y) = F_t(x,y) \cup F_1^{\omega}(x',y') \cup ... \cup .F_n^{\omega}(x',y') \cup F_1^{h}(x',y') \cup ... \cup .F_m^{h}(x',y') \tag{9}$$

Figure 7 shows the result of applying the repair-oriented feedback to one of the scene frames where a human appears with background noise. Note the extreme similarity between human pixels to be restored and their corresponding background pixels in Fig. 7.a. Taking this frame as input, an initial segmentation is done that produces the set of blobs depicted in Fig. 7.b. An analysis at object level of the blobs obtained reveals that there are several boxes associated with the human model with no pixels, which makes it possible to propose an ROI to focus upon. The normalised histograms of ROI_i and ROI_e of the parameters *angle* and *module* (Fig. 7.c-d) are analysed and the set of restored pixels associated with each parameter is obtained (Fig. 7.e-f). The joining of these pixels to the blobs already existing in the initial segmentation produces the final result shown in Fig. 7.g. As can be seen in Fig. 7.f, sometimes it is not possible to restore the

background noise from just analysing one parameter. That is why the result from analysing both parameters is accumulated.

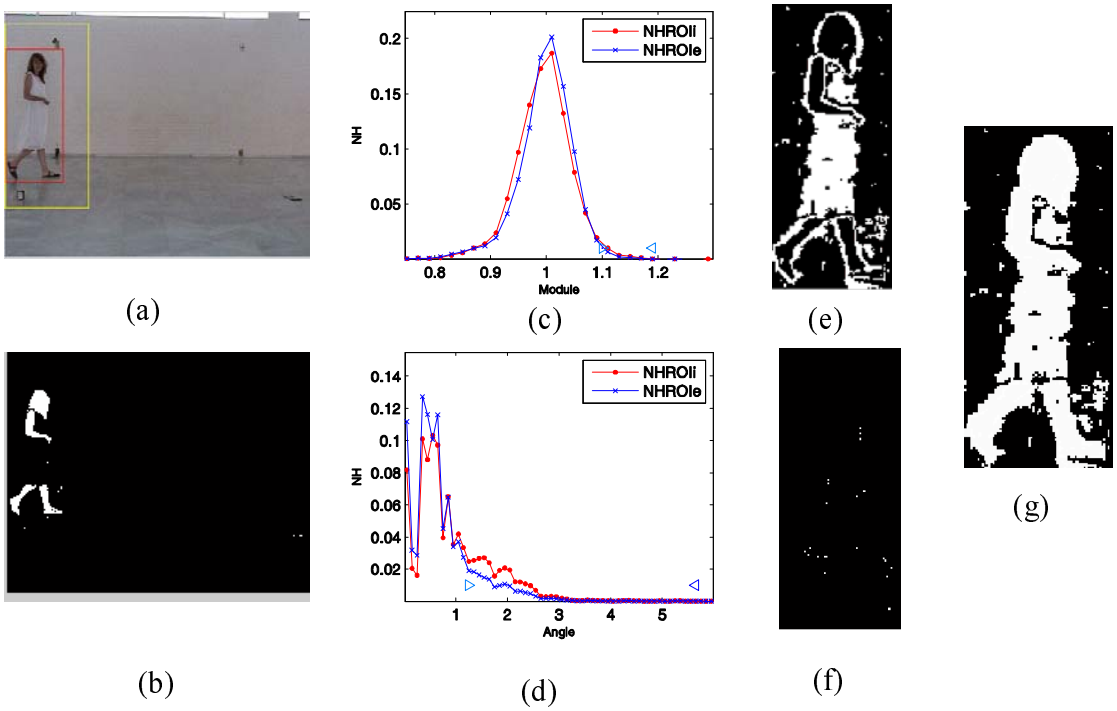

Fig. 7. Example of repair-oriented feedback loop of Fig. 2: (a) Input frame; (b) Foreground map corresponding to the initial segmentation with ROI_i (obtained in *diagnose-1*) and ROI_e.; (c) and (d) normalised histograms corresponding to parameters *w* and *h*, respectively, indicating the ranges obtained for K=1.7 (the triangles mark the beginning and end of the selected ranges); (e) and (f) pixels restored after segmenting with the new parameter configuration; (g) result of the final segmentation after feedback (joining of the initial segmentation and those obtained in (e) and (f).

Note that when we make those parts of the human that have disappeared (background noise) emerge, not only are we restoring the silhouette of the moving object, but we are also explicitly relating all those blobs belonging to the same object and which initially were not joined. This segmentation improvement mechanism could be used with any object level process which proposes an ROI hypothesis where it is assumed that an erroneous segmentation has been done, for example, after analysing a blob implying the presence of a human or when it is not possible to recognise the type of object associated with this blob.

3.2. Repair-oriented feedback between the activity and object levels

The feedback that we propose in this section is to repair the output produced by the task for detecting events, belonging to the object level. Fig. 8 shows the different tasks involved in this process, as well as the flow of information obtained from instantiating the generic scheme proposed in Fig. 2. In this instance, the task *Detect Events* identifies the primitive events that are going to be used as input to the activity level. The existence of inconsistencies between the events detected is assessed with the task *Diagnose-1* that distinguishes between normality diagnosis and error diagnosis. Normality diagnosis

indicates that no inconsistencies have been found between events; therefore, all the activity level processing can be done from these events. Inconsistencies are defined as event incompatibility patterns. If the diagnosis identifies some inconsistency, a therapy is proposed to act on those operators that determined the events participating in the inconsistency pattern, either in the sense of reconfiguring the input parameters or applying a new operator. Thus, in a second phase, the operator with the new parameter configuration or a different operator is executed. The feedback cycle is accordingly completed.

To exemplify this type of feedback, we are going to use the scenario that we have been working with [12]. It is an indoor space that is a pass-through area for humans. Humans can move freely, come in and go out of the observation area, sit down, carry a briefcase, leave it, pick it up, etc. An alarm will go off when someone leaves a briefcase (a package) in the area under surveillance and tries to abandon a larger site (airport, hospital, etc.). If someone picks up the briefcase, the alarm will stop.

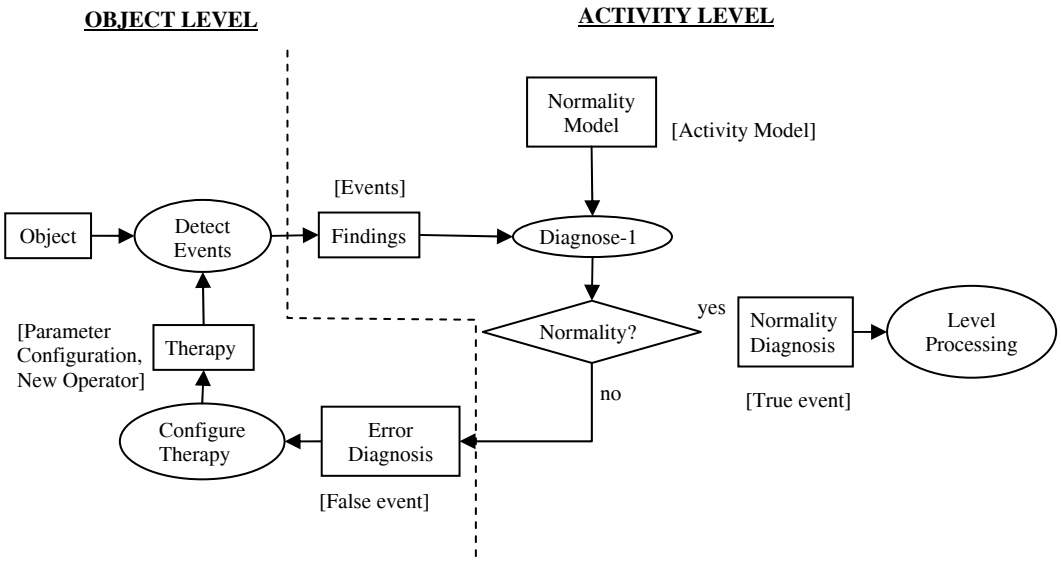

Fig. 8. Repair-oriented feedback structure for event correction.

Event detection

For the scenario described above, the primitive events that emerge at the activity level from the object level are shown in Table 1. These events are obtained using a set of operators that handle information from the object level. Movement information obtained in the tracking task is used to determine the events *Enters, Goes-Out, Stops, Begins-To-Walk, Appears, Disappears, Is-Near-To, Is-Far-From, At, Is-Going-To, Is-Going-Back* and *Goes-Out*. Other events, such as *Crouches-Down, Gets-Up, Detected-Man-Object* and *Not-Detected-Man-Object* are obtained with operators that use the human block model described in section 3.1. Some of these events are obtained by default (monitoring process) and others upon request from the activity level.

Events (from object level)	Type	Description
Enters (*h*, (*x*, *t*))	monitoring	The human *h* appears in the scene at space-time position (*x*, *t*).
Goes-Out (*h*, (*x*, *t*))	monitoring	The human *h* disappears from the scene at space-time position (*x*, *t*).
Stops (*h*, (*x*, *t*))	monitoring	The human *h* stops at space-time position (*x*, *t*).
Begins-To-Walk (*h*, (*x*, *t*))	monitoring	The human *h* begins to walk at space-time position (*x*, *t*).
Crouches-Down (*h*, (*x*, *t*))	monitoring	The human *h* crouches down at space-time position (*x*, *t*).
Gets-Up (*h*, (*x*, *t*))	monitoring	The human *h* gets up at space-time position (*x*, *t*).
Detected-Man-Object (*h*, *o*, (*x*, *t*))	monitoring	Detected that the human *h* is holding the object *o* at space-time position (*x*, *t*).
At (*h*, *x*, *t*)	monitoring	The human *h* is located at space-time position (*x*, *t*).
Not-Detected-Man-Object (*h*, (*x*, *t*))	monitoring	Detected that the human *h* is not holding any object at space-time position (*x*, *t*).
Appears (*vo*, (*x*, *t*))	monitoring	The visual object *vo* (object or human) appears at space-time position (*x*, *t*).
Disappears (*vo*, (*x*, *t*))	monitoring	The visual object *vo* (object or human) disappears at space-time position (*x*, *t*).
Is-Near-To (*h*, *d*, *t*)	upon request	The human *h* is close to door *d* at time *t*.
Is-Far-From (*h*, *d*, *t*)	upon request	The human *h* is far from the door *d* at time *t*.
Is-Going-To (*h*, *d*, *t*)	upon request	The human h is going to door *d* at time *t*.
Is-Going-Back (*h*, *d*, *t*)	upon request	The human *h* is going back from the door *d* at time *t*.
Goes-Out (*h*, *d*, *t*)	upon request	The human *h* goes out through the door *d* at time *t*.

Table 1. Simple events detected at the object level which are transmitted to the activity level.

By way of example, the operator *Operator_Detected-Man-Object* determines the events *Detected-Man-Object* and *Not-Detected-Man-Object* from the block model. This operator analyzes human morphology in specific blocks to calculate whether the human is carrying an object or not. In particular, it focuses its study on the block with greater probability of containing the object and compares the dimensions of the regions contained in this block with others for other parts of the body. We have considered two possible positions of the object for the human: situated partially in block B5 (see Figure 5), where the human is carrying a suitcase or bag, and situated partially in block B2, where the human is carrying a rucksack. In the first instance, block B5 is compared with blocks B2 and B3 (which corresponds to the human trunk), thereby making it possible to determine its width approximately. By comparing the width of the trunk with the widest region found in B5 we can determine whether the human is carrying the suitcase

or not. In Fig. 9 the difference can be seen in the region in block B5 of a human not carrying a suitcase (Fig. 9.a) with when he carries it (Fig. 9.b). To summarise, the input parameters to the operator, in this instance, will be (B5, B2, B3). In the second instance, block B2 is compared with blocks B3 and B4 where most of the human trunk is found. Now, the input parameters to the operator will be (B2, B3, B4). In accordance with the scenario posed, initially, in the monitoring process, we will assume that the object carried has a higher probability of being in block B5.

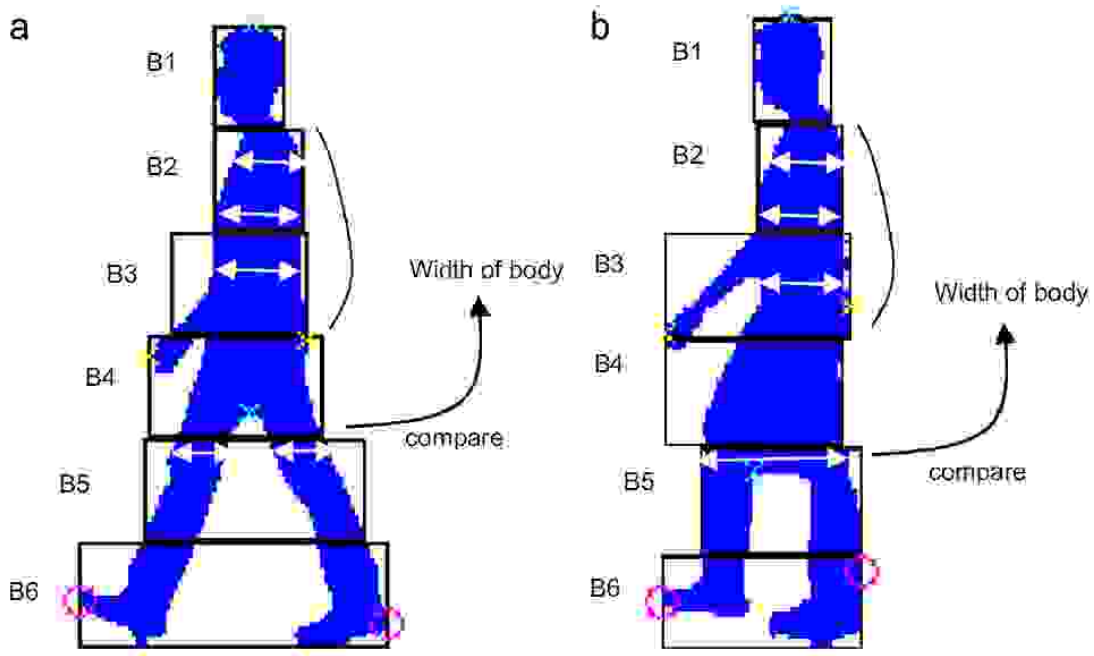

Fig. 9. Examples of a block model for a human without a suitcase (a) and a human with suitcase (b). Note the difference in width in the regions in B5 compared with the width of the trunk in B2 and B3.

Diagnosis

The set of primitive events plays the role of *findings* in the activity level and will be the input to the diagnosis task. The aim is to assess whether the set of events detected is consistent with the knowledge of the activity level.

Let us look at a specific example. At a given moment the primitive event *Appears*(o_1,(x_1,t_1)) reaches the activity level from the object level, i.e., at instant t_1 the inanimate object o_1 has appeared on the scene in position x_1. The task *Diagnose-1* must check whether any human h is near to it at instant t_1 and whether this human, at earlier instants, carried an object, in other words, the event *Detected-Man-Object*(h, o, (x,(k,...,t-1))) was active, k being the instant when the human appeared on the scene. The inconsistency could appear either because there are no humans near the object at the same instant or because although they exist, none of them carried an object at previous instants. In the first instance, it could be thought that it is an error of the event *Appears*(o_1,(x_1,t_1)), i.e., this object does not exist in the image, or, in the second instance, it is a detection error of the event *Detected-Man-Object*(h, o, (x_h,(k,...,t-1))). Anyway, feedback to correct the error is necessary.

Therapy configuration

The aim of this task is to correct the event that produced the inconsistency. Depending on the event to be corrected the therapy will differ. In the example, a first option diagnoses that the erroneous event is *Detected-man-object(h, o, (x_h,(k,…,t-1)))*. If from this assumption, feedback is unable to correct the error, a second option diagnoses that the erroneous event is *Appears(o_1,(x_1,t_1))*.

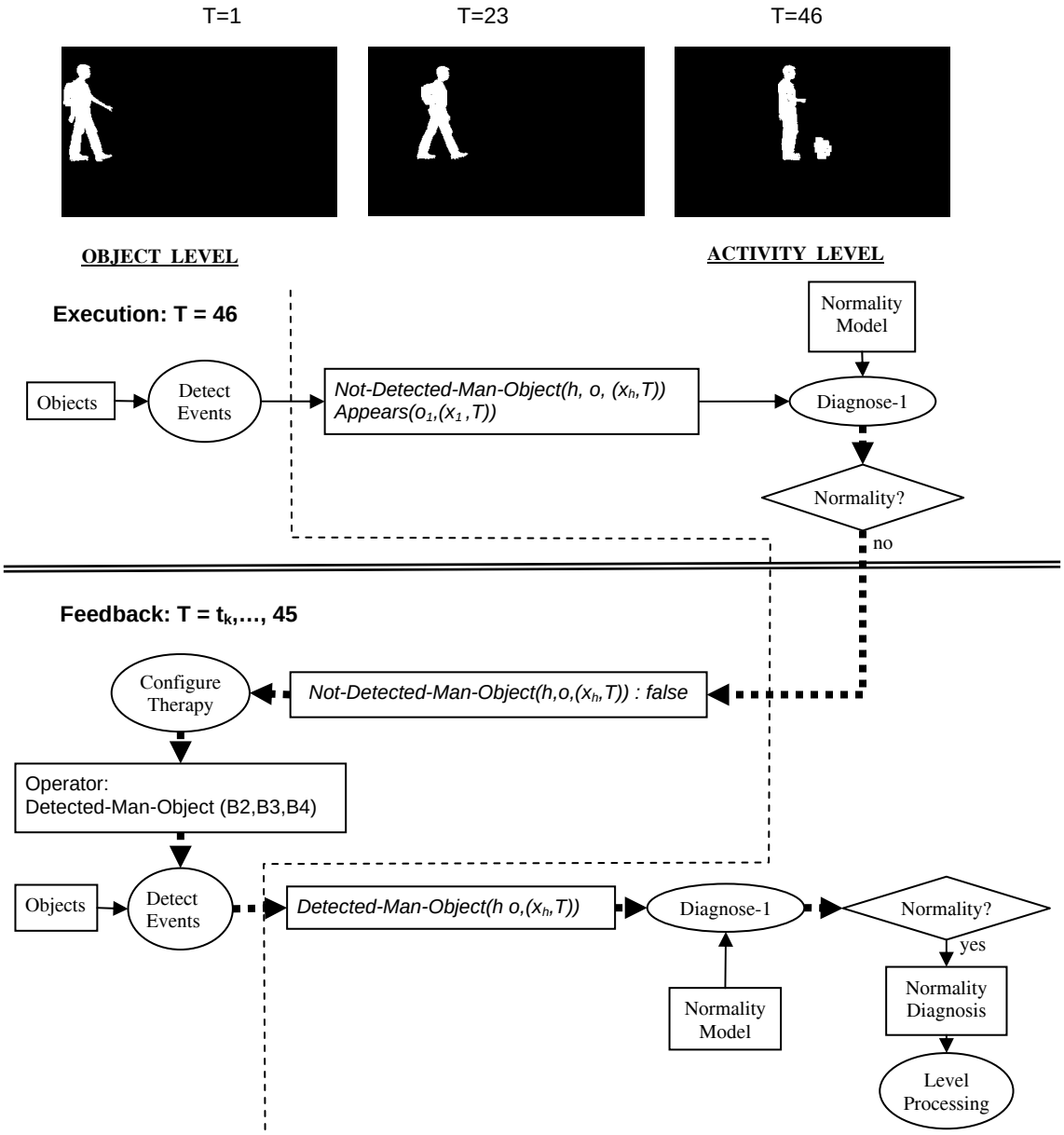

Fig. 10. Example of repair-oriented feedback for the event *Detected-man-object.*

When the erroneous event is taken to be *Detected-Man-Object(h, o, (x_h,(k,…,t-1)))*, the therapy consists of reconfiguring the input parameters to the operator *Operator_Detected-Man-Object* which become (B2, B3, B4). However, if the erroneous

event is taken to be *Appears*(o_1,(x_1,t_1)) we shall assume that it has been the effect of a reflection and this event will be deactivated.

In Fig. 10 we show how this example is executed. At instant $t_1=46$, an error occurs after diagnosis due to inconsistency between the events *Appears*(o_1,(x_1,46)) and *Not-Detected-Man-Object*(h, o, (x_h,(k,…45)). Feedback is done (broken line), configuring the input parameters to the operator *Operator_Detected-Man-Object* with the values (B2, B3, B4). Then, the operator generates the event *Detected-Man-Object* for each instant. Immediately after, *Diagnose-1* analizes the new events. Now there are no inconsistencies, since it was detected that the human was carrying a rucksack, and the activity level processing continued.

3.3 Focus-oriented feedback between the activity and object levels

For surveillance of very simple activities a *monitoring* stage is sufficient whose basic process in the activity level is event composition. In this level, as we have mentioned, from object level simple events, complex events are composed that describe significant high-level abstraction activities within the scenario considered. This composition forms part of the level processing according to the terms in the scheme in Fig. 2. An example of this type of inference: the event *Leaves()*, which is activated when it is detected that a person has left an object, is based on the occurrence of its component events and their spatial-temporal constraints (Table 2). This event is composed by instantaneous events, like *Appears*, and state-events, with a specific duration occurring since t_i to t_j, like *Carrying* and *Not-Carrying*. In this table, x_i is a location, h_i is a human and o_i is an object. The expression $t_i >> t_j$ is used meaning that t_2 is "sufficiently" greater than t_1 so that indeed the event *Not-Carrying* is significant. The expression $t_i \geq t_j$ is used meaning that t_i is equal o "lightly" greater than t_2. The symbol ≅ is also used to express the condition that the two locations are "sufficiently" near so that the composition that is based on this quasi-coincidence makes sense.

Composed Event	Components
Leaves (*h*, *o*, (*x*, *t*))	events: *Carrying (h,o1,(-,t1), (x1,t2))* and *Appears* (*o*, *x*, *t*) and *Not*-Carrying *(h,o1,(-,t1),(x1,t2))* constraints: $x_1 \cong x_2 \cong x$ $t_4 >> t_3 \geq t \geq t_2 >> t_1$

Table 2. Composition of event *Leaves.*

Yet the scenarios are usually complex, and especially so if the reaction time is critical, and it is not possible to obtain all the necessary characteristics of the images in the monitoring. It is therefore necessary to include, in addition to the monitoring-composition, a diagnosis stage of the situation that adds a top-down organisation, which corresponds to the usual concept of knowledge-guided search. This type of feedback enables the system to assign (or reassign) computational resources by generation hypothesis and the associated selective search of confirmation findings not available from the emergence.

The distinction between bottom-up emergence of higher-level abstraction data and top-down attention search is usual, for example, in clinical medicine, where symptoms and signs from lower cost and more reactive clinical exploration are distinguished from those from complementary explorations (for example, magnetic resonance scanning) where higher cost resources are assigned based on suspicion or hypothesis. Already in image interpretation, in the work of Howarth and Buxton [8] monitoring was explained as "passively passing from images to conceptual descriptions", while *watching* is "more active and task oriented", with feedback between a pre-intentional level from where behaviours emerge (from images), and an intentional control level, which focuses on the surveillance task process.

The inferential scheme in Fig. 11 shows the key subtasks for diagnosing situations as part of the activity level processing in the overall surveillance task. A *claim* resulting from the event composition leads us to pose (*generate*) hypotheses of situations (future, past or present) relevant for scenario surveillance, which are necessary to check according to the determination (*prediction*) of confirmation findings. Chleq and Thonnat [4] also include this hypothesis approach, which implies the need to explore alternative solutions in parallel. Discriminating these hypotheses will require focusing on new characteristics of the images and, therefore, requesting the participation of new visual operators in the object level. Fig. 11 also shows the connection of the roles in the inferential scheme with the domain entities, according to KADS methodology [22] for knowledge based system development. The subtask *generate* is based on the *suggestion relation* between *findings* and *anomalies*, while the subtask *predict* is based on the *confirmation relation* between *finding* and *anomaly* that would confirm them (*confirmation findings*). The assessment of the current availability of these findings corresponds to task *Diagnose-2* in the scheme in Fig. 2. Otherwise, this same task would transmit the request of these findings to the object level as "*requested findings"*, which in turn, would select new operators (*Configure Therapy*), thereby closing the feedback loop. Fig. 12 shows an example of this: the sequencing of prealarms and an alarm that makes it possible to adjust the computational cost to the danger of the activity identified. The composed event *Leaves(h, o, t)* places the system in the initial state of prealarm. From here it returns to a state of normality if an event *Picks-Up (h',o, t)* is activated on the same object (it could have a different human h' as a first argument). In this initial prealarm state a request is made to the object level to search for the event *Is-Near-To(Exit)*. If it is obtained, the system passes to a higher prealarm state. This state requests the search for *Is-Far-From(Exit)* and *Is-Going-Out*. If it receives the event *Is-Far-From(Exit)*, it returns to the previous prealarm level. If it receives *Is-Going-Out* it passes to the alarm, which implies communication to the internal or external guard.

The Fig. 13 shows the main window of the surveillance prototype that we have developed. It simultaneously describes the events at the four levels. The top left window shows the original frame of the current processing instant, while the top right window shows the segmented frame of the same instant with the superimposed information of the identify objects. In the bottom left window the fired events are represented, while in the bottom right window the composed events or activities inferred by the system are presented. Both spaces add the new inputs to the top section and use a horizontal line to separate the events of each instant of time. The figure shows the window of the prototype at an instant when an event leaves has been identified and, therefore, the

system is in the initial state of prealarm. In the last subwindow the consistent search for findings is notified (as request type events), which focuses on the object level events that would enable the state to evolve, either towards normality or towards the following prealarm state.

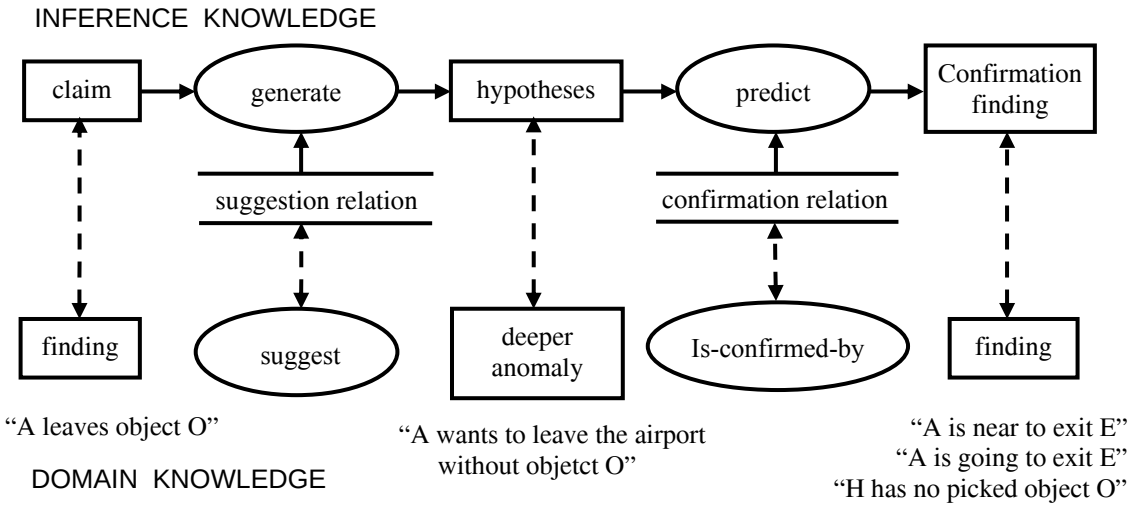

Fig. 11. Inferential structure of the situation diagnosis forming part of the activity level processing, which generates hypotheses and predicts the findings confirming them. Inference knowledge linking to domain knowledge is also shown.

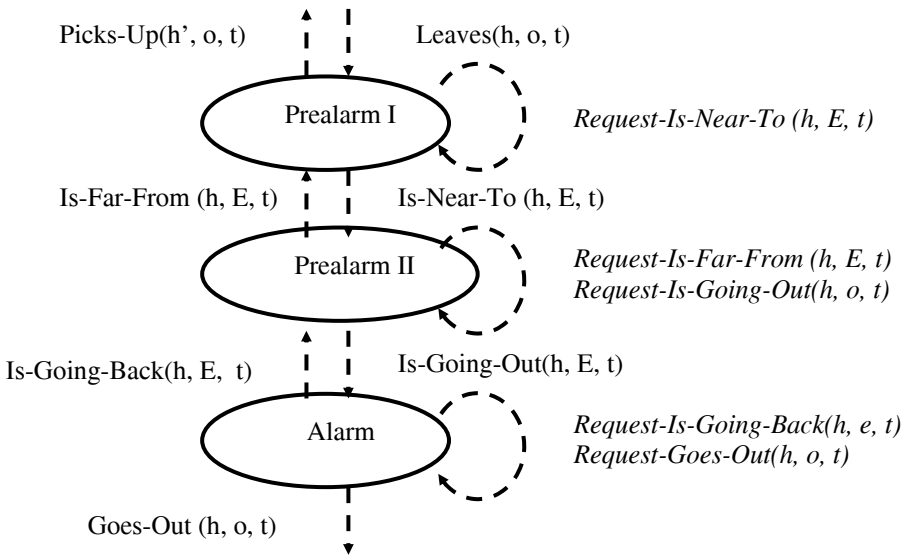

Fig. 12. Diagram of states representing the transitions of *prealarm* states until the very *alarm* goes off. *Request* type events are sent from the activity level to the object level, focusing on the search for new findings. Output events are written in cursive letter.

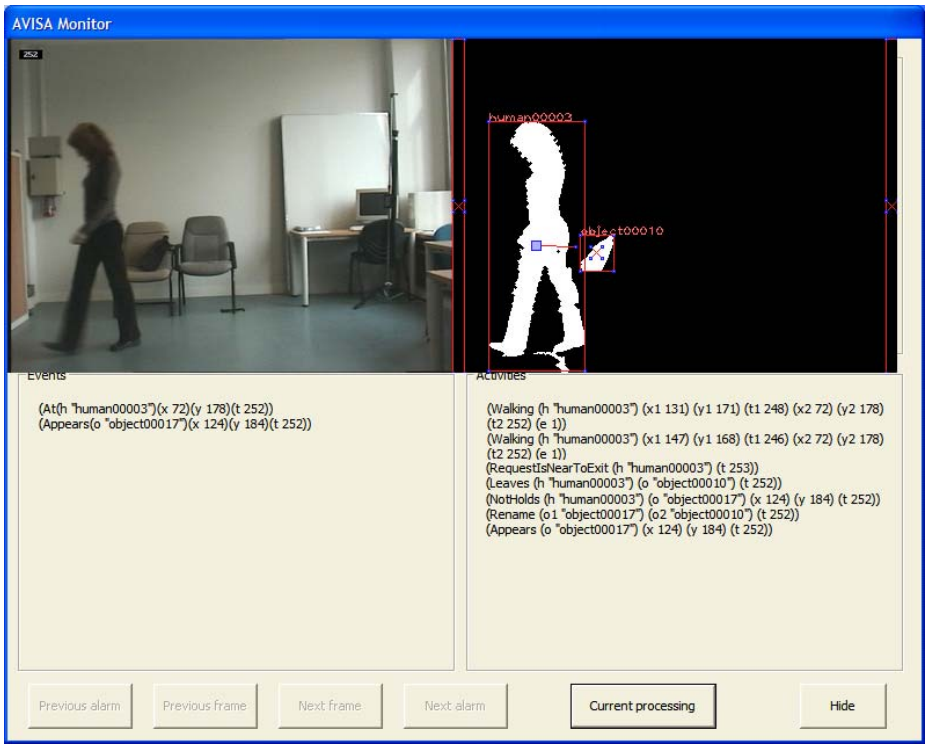

Fig. 13. Main window of the surveillance prototype with which it is possible to monitor what is happening in any of the four levels: the raw image from the camera (top-left), the segmented and identified objects (top-right), the events in the object level as a result of identification and tracking (bottom-left) and, finally, the resulting composition of events in the activity level (bottom-right).

4. Conclusions

In this work we have proposed a general top-down feedback scheme between adjacent description levels to improve the interpretation of video sequences. This scheme distinguishes two types of feedback: repair-oriented feedback and focus-oriented feedback. It highlights the improvement from using information from higher semantic levels in the sense of "repairing" errors and "focusing" resources to locate findings that with just bottom-up organisation is impossible, unless very superior computational resources are available.

Three feedback examples of both types are shown for interpreting different video sequences in surveillance applications: the first improves segmentation in the blob level from object level information, the second resolves inconsistencies in object level events from activity level knowledge, and the third adjusts the computational load according to the degree of alarm of the events detected in each specific surveillance scenario.

A visual surveillance prototype was implemented that integrates bottom-up with top-down organisation according to this generic feedback scheme. This prototype shows the scene representation in the four description levels simultaneously. Thus, it is possible to compare the results before and after introducing the aforementioned feedback scheme, thereby improving the identification and description of the scene events of interest and the system's general performance.

Acknowledgements

The authors are grateful to the CICYT for financial aid on project TIN-2004-07661-C0201 and the UNED project call 2006

References

[1] A. Bobick. Movement, Activity, and Action: The role of knowledge in the perception of motion. Royal Society Workshop on Knowledge-based Vision in Man and Machine, London, (1997), pp. 1257-1265

[2] H. Buxton and S. Gong. Advanced visual surveillance using bayesian networks. In Procedings of the Fifth International Conference on Computer Vision (ICCV 95), June 20-23, 1995, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA. IEEE Computer Society, (1995), pp. 111-123.

[3] E.J. Carmona, J. Martínez-Cantos and J. Mira. A new video segmentation method of moving objects based on blob-level knowledge, Pattern Recognition Letters 29, (2008), pp. 272-285.

[4] N. Chleq and M. Thonnat. Realtime image sequence interpretation for video-surveillance applications. IEEE Int. Conf. On Image Processing (ICIP'96), Laussane, (1996), pp. 800-804.

[5] E. Folgado, M. Rincón, E.J. Carmona and M. Bachiller. A block-based model for monitoring of human activity. Technical Report AVISA-12-07, (2007).

[6] S. Hongeng, R. Nevatia and F. Brémond. Video-based event recognition: activity representation and probabilistic recognition methods. Computer Vision and Image Understanding 96, (2004), pp. 129-162.

[7] R.J. Howarth and J. Richard. Interpreting a Dynamic and Uncertain World: High-Level Vision. Artif. Intell. Rev. 9:1, (1995), pp. 37-63.

[8] R.J. Howarth and H. Buxton. Conceptual descriptions from monitoring and watching image sequences. Image and Vision Computing 18, (2000), pp. 105-135.

[9] Y. Ivanov, C. Stauffer, A. Bobick and W.E.L. Grimson. Video surveillance of interactions. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'99) - 2nd. Int. Workshop on Visual Surveillance. Fort Collings, Colorado, (1999), pp. 82-89.

[10] R. Martínez-Tomás, M. Rincón, M.Bachiller and J. Mira. On the correspondence between Objects and Events for the Diagnosis of Situations in Visual Surveillance Tasks, Pattern Recognition Letters 29(8), (2008), pp. 1117-1135.

[11] J. Mira. Memoria del proyecto TIN2004-07661-C0201, AVISA: Diseño e implementación de un conjunto de agentes de diagnóstico, planificación y control, con capacidad de aprendizaje y cooperación con humanos en tareas de vigilancia, UNED, 2004.

[12] J. Mira, R. Martínez-Tomás, M. Rincón, M. Bachiller and A. Fernández-Caballero. Towards a semi-automatic situation diagnosis system in surveillance tasks based. In J. Mira & J.R. Álvarez (eds.) Nature Inspited Problem-Solving Methods in Knowledge Engineering (IWINAC- 2007), LNCS-4528, Springer-Verlag, (2007), pp. 90-98.

[13] R. Moeller, B. Neumann and M. Wessel. Towards computer vision with description logics: Some Recent Progress. Proc. Speech and Image Understanding, IEEE Computer Society, (1999), pp. 101-116.

[14] H.H. Nagel. From imagine sequences towards conceptual descriptions. Image and Vision Computing 6:2 (1988), pp. 59-74.

[15] H.H. Nagel. Steps towards a cognitive vision system. AI Magazine 25:2 (2004), pp. 31-50.

[16] B. Neumann and H. Novak. Events models for recognition and natural language descripcion of events in real-world image sequences. Proceedings of the Eighth IJCAI , Karlsruhe. Morgan Kaufmann, San Mateo, Calif., (1983), pp. 724-726.

[17] B. Neuman and T. Weiss. Navigating through logic-based scene models for high-level scene interpretations. 3rd International Conference on Computer Vision Systems, ICVS-2003. Lecture Notes in Computer Science 2626, (2003), pp. 212-222.

[18] B. Neumann. Natural language description of time-varying scenes. Brericht no. 105, FBI-HH-B-105/84, Fachberic Informatik, University of Hamburg, (1984).

[19] C.S. Pinhanez and A.F. Bobick. PNF propagation and the detection of actions described by temporal intervals. DARPA Image Understanding Workshop, New Orleans, Lousiana, (1997), pp 227-234.

[20] M. Rincón, E.J. Carmona, M. Bachiller and E., Folgado. Segmentation of moving objects with information feedback between description levels. J.Mira and J.R. Álvarez (Eds.) IWINAC 2007, Part II, LNCS 4528, (2007), pp. 171-181

[21] N.A. Rota and M. Thonnat. Video sequence interpretation for visual surveillance. In IEEE Int. Workshop on Visual Surveillance. (VS'00). Dublin, Ireland, (2000), pp 59-68.

[22] G. Schreiber. Knowledge Engineering and Management: The Commonkads Methodology. MIT Press, (2000).

[23] J.M. Siskind. Reconstructing force-dynamic models from video sequences. Artificial Intelligence 151 (2003), pp. 91-154.