Video Sequence Interpretation for Visual Surveillance

Nathanael Rota
INRIA Sophia Antipolis, France
Nathanael.Rota@sophia.inria.fr

Monique Thonnat
INRIA Sophia Antipolis, France
Monique.Thonnat@sophia.inria.fr

Abstract

This paper presents recent work done on video sequence interpretation. We propose a framework based on two kinds of a priori knowledge : predefined scenarios and contextual information. This approach has been applied on video sequences of the AVS-PV visual surveillance european project.

1. Introduction

This paper presents recent work we have done on video understanding [1]. Our goal is to detect mobile objects (specially people) and to analyze their behavior. We propose a framework based on a priori knowledge. This work is based on three hypotheses: first we consider a static camera, second we use a unique monocular camera and third we deal with real-time constraints. The first hypothesis (static camera) is often verified for current visual surveillance networks and allows us to simplify the low-level detection of mobile objects w.r.t. a fixed environment. The second hypothesis (unique monocular camera) is verified in almost all current visual surveillance networks. The third hypothesis (real-time constraints) is very interesting as it implies that the solutions should be kept with a minimal computing time. But it also implies a fully automated system. After a presentation of related work (Section 2), we present (Section 3) the current low-level image processing techniques used for mobile object detection and tracking. In Section 4 we describe the role of a priori contextual information and different ways of representing this information. Then (Section 5) we adress the problem of high-level description of mobile object behavior using generic observable events and application-dependent scenarios. Finally, (Section 6) results obtained on different visual surveillance applications in the european Esprit project AVS-PV are shown and discussed. The paper concludes with future work for enhancing the robustness of such image understanding systems.

[1]Work done with support from Dyade GIE Bull INRIA

2. Related work

Cohen, Bremond Medioni and Nevatia (University of South California), in DARPA's VSAM focus on event recognition involving vehicles and humans ([14] and [7]). The particularity of this work is that videos are filmed by non-fixed camera. They used models of maps of the environment to place aerial images in an *a priori* known map. They used a property net to compute events and states, which controls the evolution of predefined automaton describing situations. Herzog (VITRA) proposes a system able to dynamically describe scene with humans. The originality of his work is the application environment: a soccer stadium ([1] and [10]) and the inference method based on time interval logic, to describe temporal sequence of events, which are computed and typed separatly. Intille and Bobick (MIT Media lab), in a similar environment, focus on analysis of American football scenes. Their aim is the recognition of particular strategies in the complex players' interactions ([11] and [12]). The main point is that those activities are not just human behaviors but human group behavior. Shah (University of Central Florida) is interested by dynamic description of human behaviors in office environments ([2] and [8]). Even if the problem is the recognition of long duration activities, the authors insist on the importance of the recognition of 'key instants' which are the conditions of changing states in an automaton representing the global behaviors. The "Key instants" are generated when certain conditions are realised.

Tessier (ONERA), in the PERCEPTION project, proposes an original method to describe behavior. Petri nets are used to represent dynamic evolutions of a car park scene with humans and vehicles ([16] and [4]). Buxton and Gong (University of Sussex) gave an important contribution to the domain with the VIEWS project ([3]). The system was able to deal with humans and vehicles on roads, streets or in car parks. A high level representation based on Bayesian networks was computed. This work points out the necessity to deal with uncertainty and to use contextual information to enhance detection and tracking results. In the same vein, Ivanov and Grimson (MIT) work on detection of human and

vehicle behaviors in car park. The interest of this research is in the event's combination method ([13]). A behavior is represented by a set of rules based with a stochastic context-free grammar, which allows certain combinations of simple constant predicates. The general scheme of our approach is based on the use of predefined scenarios [6] and a priori contextual information [5]. In this paper we propose a method based on both n-ary tree to declare events [17] and temporal logics to declare application dependent scenarios.

3. Perception

In this section we will shortly present the perception component we have used. A more detailed description can be found in [15]. The perception methods are standard ones which verify the real-time constraint hypothesis. Their role is to incrementally provide a history of the persons who have been detected in the scene. It is composed of four main sub-parts: motion detection, person detection, person tracking and smoothing. Each sub-part contains alternative methods which are manually selected and parametrized in a configuration phase.

3.1. Motion detection

Motion detection is basically a thresholding of the difference of the current image with respect to a reference image. Before the difference is computed we filter the current image with a 3×3 gaussian filter to reduce noise. Then for each pixel we compute the absolute difference between its intensity (grey or colour) and the intensity of the corresponding pixel in the reference image. If this difference is greater than a certain threshold α, the pixel is marked as mobile and otherwise it's marked as stationary. We then update the reference image I_r with information from the current image I_c according to the following equation:

$$I_r = (1 - \beta)\, I_r + \beta I_c$$

We see that for $\beta = 1$ we detect motion as the difference between subsequent images in the sequence and that for $\beta = 0$ motion is detected with respect to a fixed background image. α and β are parameters of the motion detector.

3.2. People detection

The goal is to detect which mobile regions (blobs) correspond to a person. We use a model of a person with 8 parameters: the position of the center of gravity (px_{img}, py_{img}), the height h_{img} and the width l_{img} in the $2D$ image, the $3D$ position $(px_{3D},\ py_{3D})$ on the ground plane of the scene, the $3D$ width l_{3D} and the $3D$ size h_{3D}. The bounding box of a person is defined by the image measures (px_{img}, py_{img}), h_{img} et l_{img}. The people detection algorithm analyzes the set of blobs. Both $2D$ image criteria and $3D$ scene criteria are used. The first ones are based on the $2D$ distance between blobs in the image. The goal is to merge the closest blobs in the image. The second ones are based on constraints on the $3D$ height and width. The $3D$ measures are obtained by linear projection of the image plane.

3.3. People tracking

The goal of this step is to update the set of trajectories. For that purpose, the persons who have been detected in the current image must be matched with those detected in the previous ones. This matching can be defined as a function from the set P_{t-1} of persons detected a time $t-1$ into the set P_t of the persons detected a time t. We use 3 alternative methods: a method based on the amount of overlap in the $2D$ image, a method based on the proximity of the persons in the $3D$ scene and a restrictive method based on the proximity of the persons in the $3D$ scene. The first method (based on the amount of overlap in the $2D$ image) states that two persons detected at two consecutive times are the same real person if the percentage of overlap of their bounding box is greater than a threshold. The second method matches a person at time t with a person at time $t-1$ if their $3D$ distance is below a threshold. The third method is similar to the second one, but the function must be either an injection or a surjection.

3.4. Smoothing

The first goal of this step is to correct errors made in the previous perception steps on the different $3D$ measures of a person: $(px_{3D},\ py_{3D})$ the position on the ground plane, h_{img} the height and l_{img} the width. The second goal is to estimate $(vx_{3D},\ vy_{3D})$ the instantaneous speed of the persons. Three smoothing methods are used. The first method uses a standard Kalman filter. The state vector is defined by $(px_{3D},\ py_{3D},\ vx_{3D},\ vy_{3D})$. The linear dynamic model is based on the hypothesis of a constant speed. The second and third methods are respectively median and mean filtering with temporal window size 3, 5 or 7 . $(vx_{3D},\ vy_{3D})$ is initialized by computing $v(t) = \frac{p(t)-p(t-1)}{\delta t}$ then each of the four values px_{3D}, py_{3D}, vx_{3D} and vy_{3D} are filtered.

4. Contextual information

As our goal is to provide a framework which can be adapted to specific conditions we propose to define two kinds of a priori informations : contextual information (see this section) and predefined scenarios (see section 5). Contextual information is an a priori information which con-

tains a description of the static environment observed by a camera. For each particular camera looking at a particular environment, a security operator must provide, in a configuration phase, pertinent information according to the formalism we propose. For more details on the role of context in video understanding see [5]. The context contains in addition to geometric information some semantic information. Its structure is made of a set physical objects, a set of interesting areas and a calibration matrix which enables the passage from the 2D image plane to 3D coordinates.

The geometry of the interesting areas is described by a list of polygons defined in planes which may have any orientation. The geometry of each physical object, or piece of equipment, is a generalized cylinder defined by its height and its polygonal basis.

The semantic information of each piece of equipment and of each interesting area is made of six attributes : four with symbolic values and two with numerical values. The four attributes with symbolic values are : the type (equipment or area), the function (i.e. table, seat, corridor, etc...) the name (i.e. seat3, corridor2, etc...) and characteristics (i.e. yellow, fragile, etc...). The two attributes with numerical values which are very useful for scenario recognition are: the normal distance and the normal time of usage of an equipment.

Figures 1 and 2 show two examples of environments we have modeled; for each example the left image shows the view observed from the camera and the right image shows the 3D model of the same environment based on the geometrical information contained in the context. For instance for the first example of a coffee room the context contains: the calibration matrix, the description of nine pieces of equipment (three seats, one table, one coffee machine, one elevator, one dustbin, one door, and one heater) and the description of three areas (a seat area, an entrance, and a corridor). The second example is a real scene of metro station in Nuremberg which has been selected in the european project AVS-PV. This is an entrance of a metro station, eight equipments and two areas have been defined. The equipment are six turnstiles and two ticket vending machines. The areas are an entrance and a corridor.

Figure 3 shows an example of the complete description of a ticket vending machine for a metro station in Nuremberg.

5. Interpretation

In this section we adress the problem of high-level description of mobile object behavior using generic observable events and application-dependent scenarios. The recognition process of temporal concepts can be reduced to the recognition of atemporal ones: object states. An event is a spatio-temporal property which represents a significative

Figure 1. Left: image of a coffee room. Right: $3D$ scene model.

Figure 2. Left: image of a metro station. Right: $3D$ scene model.

name =	ticket vending machine 1
type =	equipment
function =	ticket vending machine
characteritics =	fragile
proximity =	100 cm
normal time =	30 s
polygon =	([0, 415, 0],[0, 520, 0,], [-50, 520, 0], [-50, 415, 0])
height =	180 cm

Figure 3. Example of description of a ticket vending machine

change in the state of objects in the scene. Typical events usally are “to enter”, “to start running”, “to stand up” or “to leave”. The algorithm for event recognition is the following : an event is recognized if for a given object state, the value of this state is significantly different between an image (corresponding to time t_0) and another image (corresponding to time t_n, with $t_n = t_0 + d_{rec}$. The time interval between I_0 and I_n is called recognition delay d_{rec}.

For instance, if a person is, at time t_0, far from a coffee machine then close to that coffee machine at time t_n, then the event “the person moves close to the coffee machine” is recognized.

The problem of event recognition can be reduced to the atemporal problem of finding a set of states describing the scene with enough accuracy. In other words, solving the problem of event recognition is reduced to solving the problem of symbolic description of the scene. So, if for each image we have a symbolic description of the scene, it is sufficient to compare these descriptions to know the changes which had happened, i.e. the events which have occured. This scene description must perform the passage from numerical to symbolic values and must be generic enough to be applied to different environments and different applicative domains.

5.1. State model

The objective is to provide a set of generic states based on a formalism which enables its extension and its parametrization. A state of objects in the scene is defined by an n-ary tree which represents the way this state is computed (see Fig. 4 an abstract example of such a tree). Four types of nodes are distinguished: object nodes, descriptor nodes, operator nodes and classifier nodes (see below for their definition). The leaves of this tree are the objects involved in that state. Father nodes of the leaves are numerical descriptors of these objects. All intermediate nodes are operator nodes. The root node is always a classifier node which computes the symbolic value of the object state. The minimal tree structure is reduced to 3 nodes, 1 object leaf node 1 descriptor intermediate node and 1 classifier root node. The number of branches of the tree and the length of the branches are free.

Objects. Objects are the objects of the scene at time t, i.e. an element of O, the set of the objects $o_{i,j}$ where i is the class of the object and j its label. For instance the object $o_{person,1}$ is a mobile object which has been recognized as being a person and whose label is 1. $o_{equipment,door2}$ is an object belonging to the class equipment labeled as door2.

Descriptors. Descriptors are functions defined from O to R^p to access an object measure. For instance, the size, the position, the shape, the trajectory, the orientation or the volume are possible descriptors. This notion ensures the anchoring of the model in the numerical results of the perception component.

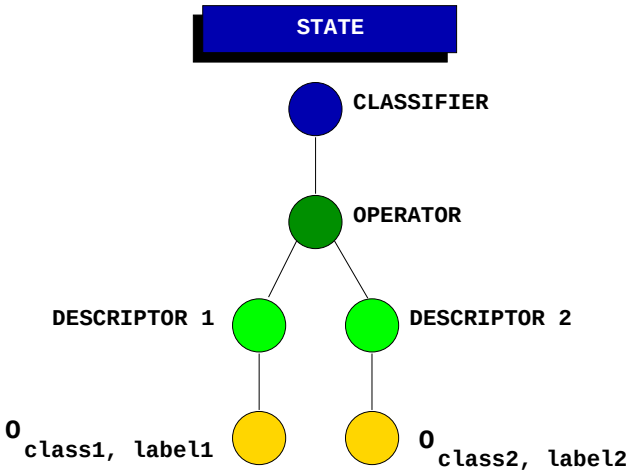

Figure 4. Example of state modeling. Objects are in light grey, descriptors are in grey, the operator is in dark grey, and the classifier in black.

Operators.Operators are functions defined from $(R^{p_1} \times \ldots \times R^{p_n})$ to R^q in order to operate on the measures. Examples of operators are the distance, the norm, the classical arithmetic or logic ones.

Classifiers. Classifiers are functions defined from R^p to S, the set of authorized symbols of the state. For instance $close$ and far can be possible symbols for a state. These classifiers ensure the passage from numbers to symbols by defining a numerical domain of definition for each symbolic value.

We have used this model of state to define a first set of states (see two examples on figure 5). For that we have defined three classes of objects, four descriptors, four operators and eight classifiers.

The 3 classes of objects are $person$, $area$, and $equipment$. Persons are the mobile objects of the scene which have been detected by the perception component. Persons are described by a vector $(px_{3D},\ py_{3D})$ representing the location of the person on the ground, a vector $(vx_{3D},\ vy_{3D})$ representing the speed vector of the person and the size h_{3D} of that person. Areas and equipments are those which are defined in the context (see Section 4). An area is a static object representing a subpart of the ground of the scene with a polygon. An equipment represents any volumic object of the environment for which we know the polygonal basis and the height h.

The 4 descriptors are: $position$, $size$, $speed$ and $shape$. More precisely: $position(o_{i,j})$ applied to a person gives

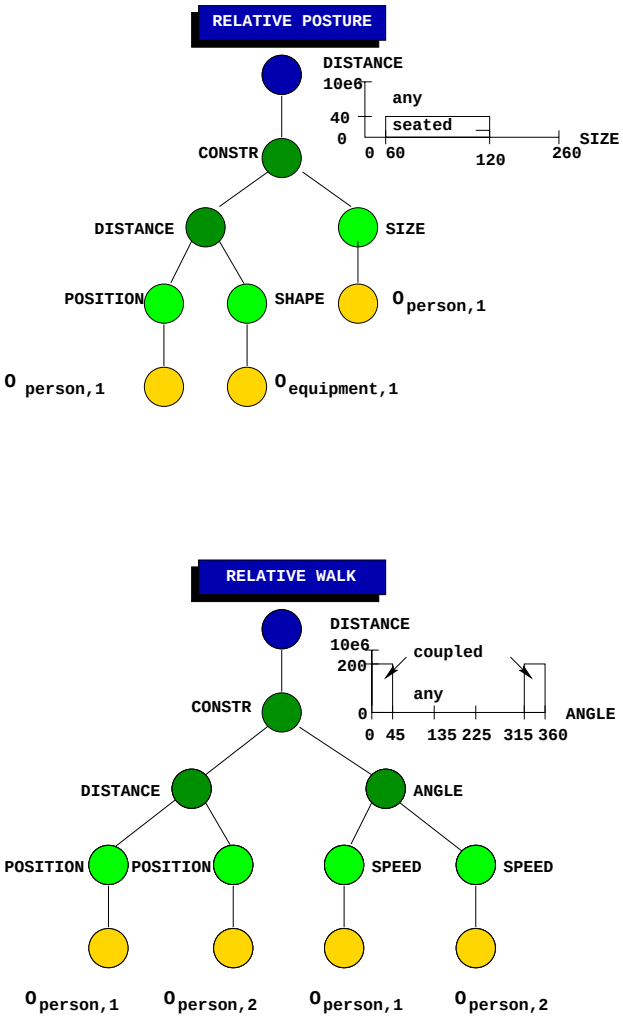

Figure 5. Two instances of the model of state. Objects are in light grey, descriptors are in grey, the operator is in dark grey, and the classifier in black.

access to $(px_{3D},\ py_{3D})$, $size(o_{i,j})$ applied to a person or to an equipment enables us to recover its size, $speed(o_{i,j})$ applied to a person returns the speed vector $(vx_{3D},\ vy_{3D})$ and $shape(o_{i,j})$ applied to an equipment or an area returns its associated polygon.

The 4 operators are: $distance$ the euclidean distance, $norm$ the norm of a vector, $angle$ the angle between two vectors in degrees and $constr$ an operator which constructs a $2D$ vector with its scalar components.

We have defined 8 classifiers which compute 8 states: $posture$, $direction$, $velocity$, $location$, $proximity$, $relative\ location$, $relative\ posture$ and $relative\ walk$. For instance we have defined (see figure 5) the state $relative\ walk(o_{person,i}, o_{person,j})$ by measuring the angle between the speed vectors of $o_{person,i}$ and $o_{person,j}$ and the distance between these two persons. If the speed vectors have a similar orientation (an angle below 45 degrees or greater than 315 degrees) and if the distance is small (below 200cm) then these persons are considered as having a $coupled$ relative walk.

5.2. Event recognition

Event recognition is now straight forward: for each image frame the object states are computed with the current objects detected in the scene at that time. If for a detected object and for a state model, there is a change in its symbolic value a new event is created at that time.

The 8 predefined state models enable us to define 18 types of event.

Posture changes create the events $o_{person,i}$ **falls down** or **crouches down** or **stands up**.

Direction changes create the events $o_{person,i}$ **goes right side** or **goes left side**, or **goes away** or **arrives**. Velocity changes create the events $o_{person,i}$ **stops** or **walks** or **starts runing**.

Location changes create the events $o_{person,i}$ **leaves** or **enters** $o_{area,j}$.

Proximity changes create the events $o_{person,i}$ **moves close to** or **moves away from** $o_{equipment,j}$.

Relative location changes create the events $o_{person,i}$ **moves close to** or **moves away from** $o_{person,j}$.

Relative posture changes create the event $o_{person,i}$ **sits on** $o_{equipment,j}$.

Relative walk changes create the event $o_{person,i}$ and $o_{person,j}$ **walk together**.

5.3. Scenario recognition

The final problem is to incrementally recognize predefined scenarios representing behaviors. A scenario is an interdependent set of events. To recognize a scenario implies to recognize all the events which compose it and to verify

their dependencies. The constraints can be temporal, spatial, logical or algebraic.

We will now give details of the scenario model we use. A scenario $s_{i,t}$, where i is the scenario identifier and t the current time of recognition, is composed of four parts: Events, Constraints, Conditions, and Success. Examples of scenario are shown in next section (see Fig. 6, 7, 11 and 12).

Events. They are the events $\{e_1, ..., e_i, ..., e_n\}$ requested by the scenario. Each event e_i is associated with the variable t_i which represents the time when e_i occured. There are two categories of events in this part: positive events and negative events. Positive events must occur for the total recognition and negative events must not occur during scenario recognition.

Constraints. They are temporal constraints $\{c_1, ..., c_i, ..., c_m\}$. Those constraints are described as first degree linear inequations on $t_1, ..., t_i, ..., t_n$.

Conditions. They are non-temporal constraints $\{k_1, ..., k_i, ..., k_p\}$ on the objects involved in the events. It forces an event object attribute to a predefined value. This attribute can be symbolic (name, function, etc...) or numerical (height, size, velocity, etc...).

Success. They are keywords $\{f_1, ..., f_i, ..., f_q\}$, which indicate the kind of feedback associated with the scenario. This part is used when the scenario is totally instanciated. There are two kinds of feedback: external and internal. External feedback is used to trigger an alarm to the security operators and internal feedback is used to generate an event to signify that the scenario has been totally recognized.

A scenario can be totally recognized, when all the events are recognized and all the constraints are verified; it can be partially recognized, when a subset S of all events are recognized and the constraints involving events of S are verified; when no event of a scenario are recognized, this scenario is called a blank scenario. The principle of the scenario recognition algorithm [6] consists of two points: as previously described, we generate, image after image, interesting events which happened in the scene, then with those events we instantiate in parallee predefined scenario models. It means that scenario recognition corresponds to updating a set of partially recognized scenarios. This scenario recognition method is an extension of the work on chronicles explained in [9].

In short, given a set of scenarios $\{s_{1,t-1}, ..., s_{i,t-1}, s_{i+1,0}, .., s_{k,0}\}$ composed of partially recognized at $t-1$ scenarios and blank scenarios and a set of events $\{e_{1,t}, ..., e_{n,t}\}$ recognized at t, the principle of scenario recognition is based on two points.

For each $s_j \in \{s_{1,t-1}, ..., s_{i,t-1}, s_{i+1,0}, .., s_{k,0}\}$, for each event of s_j if the event matches an event $e_{1,t}, ..., e_{n,t}$ and verifies the temporal constraints $c_1, ..., c_i, ..., c_m$ and the non-temporal constraints $k_1, ..., k_i, ..., k_p$, we create $s_{j,t}$. It results a new set $\{s_{1,t}, ..., s_{l,t}\}$ of scenarios. In this context, an event of s_j matches $e_{1,t}$ means that $e_{1,t}$ is an instance of this event.

We remove invalid scenarios $s_{i,t}$ from $\{s_{1,t}, ..., s_{l,t}\}$, if:
-one negative event e_k of $s_{i,t}$ has been instanciated,
-one of the $c_1, ..., c_i, ..., c_m$ implies that $s_{i,t}$ will not be instanciated.

Scenario
Name = "forbidden access to area",
Events = (t_1, $enters(p_1 : Person,\ a_1 : Area)$),
not(t_2, $leaves(p_1 : Person,\ a_1 : Area)$),
Constraints = $t_1 \leq t_2,\ t_2 \leq t_1 + 1.0$,
Conditions = function(a_1, "forbidden_access"),
Success = alarm(p_1, "has entered area",a_1)

Figure 6. AVS-PV scenario model:"forbidden access to area"

Scenario
Name = "graffiti on wall",
Events =
($t_1, moves\ close\ to\ (p_1 : Person,\ e_1 : Equipment)$),
not(t_2, $moves\ away\ from\ (p_1 : Person,\ e_1 : Equipment)$),
Constraints = $t_1 \leq t_2$,
$t_2 \leq t_1$ + normal_presence_time(e_1),
Conditions = function(e_1, "wall"),
Success = alarm(p_1, "doing graffiti onto",e_1)

Figure 7. AVS-PV scenario model: "graffiti on wall"

6. Results of metro station applications

In this part, we will describe the results obtained on real visual surveillance applications for metro stations. The videos come from the CCTV networks of the metro operators partners of the AVS-PV european project. We have formalized the expertise of three security engineers in a knowledge base containing currently 15 scenarios.

Metro Station in Brussels. In the following, we detail how two scenarios described in figures 6 and 7 are recognized. These scenarios belong to the knowledge base built for AVS-PV european project and videos have been recorded in a STIB Metro Station in Brussels. The camera observes the platform of a metro station. The aims of those two scenarios are: to prevent vandalism against equipment and to ensure the safety of passengers.

At t_1, a detected person (named Person 1 in the following) goes $inside$ the tracks area. The event "person 1 $enters$ tracks area is triggered. This area is labelled as a forbbiden area, so the first event of "forbidden access to area" scenario is recognized. Several frames after, at t_2 (see

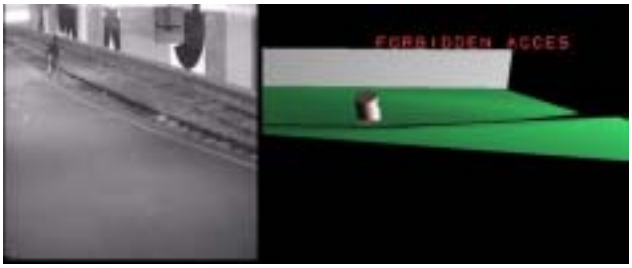

Figure 8. Left: Metro platform in Brussels at t_2. Right: 3D position of the detected person (represented by a cylinder) w.r.t the context. An alarm 'forbidden access to area" is sent to the security operator.

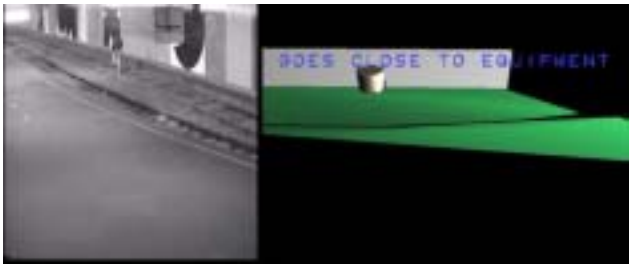

Figure 9. Left: Metro platform in Brussels at t_3. Right: an event "person 1 $moves\ close\ to$ equipment wall" is detected.

Figure 10. Left: Metro platform in Brussels at t_4. Right: an alarm 'graffiti on wall" is sent to the security operator.

Scenario

Name = "Presence period near fragile equipment",
Events =
$(t_1, moves\ close\ to\ (p_1 : Person,\ e_1 : Equipment))$,
not($t_2,\ moves\ away\ from\ (p_1 : Person,\ e_1 : Equipment)$),
$(t_3,\ stops(p_1 : Person))$,
Constraints =
$t_1 \leq t_2,\ t_1 \leq t_3,$
$t_2 \leq t_1$ + normal_presence_time(e_1),
Conditions = function(e_1, "fragile"),
Success =
alarm("Presence period near equipment",e_1),
loopback(t_2, presence_period_near_fragile, e_1, p_1)

Figure 11. AVS-PV scenario model: "Presence period near fragile equipment"

Fig. 8), Person 1 is still $inside$ the "tracks" area, so the event "Person 1 $exits$ the "tracks" area" has not been triggered. The non-occurence of the event matches the second event (negative event) of the scenario "forbidden access to area". An alarm is sent to the security operator.

Further at t_3 (see Fig. 9), Person 1 is $close\ to$ the equipment "wall", so the event "person 1 $moves\ close\ to$ equipment wall" is triggered. This event instanciates the first event of the scenario "graffiti on wall".

Further at t_4 (see Fig.10), Person 1 is still $close\ to$ the equipment "wall", so the event "person 1 $moves\ off$ equipment wall" has not been triggered. The non-occurence of the event matches the second event (negative event) of the scenario "graffiti on wall". An alarm is sent to the security operator.

Metro Station in Nuremberg. In the following, we detail how two other scenarios described in figures 11 and 12 are recognized. These scenarios belong also to the knowledge base built for AVS-PV european project and videos have been taken in a VAG Metro Station in Nuremberg (Germany). In this example, the camera observes the entrance of a metro station. The aim of those two scenarios is to prevent vandalism against ticket vending machines. These machines have been defined in the context (see Section 3) as fragile equipment.

At t_1, a detected person (named Person 1 in the folllowing) is far from an equipment labeled as "fragile". Further at t_2 (see Fig. 13), person 1 is $close$ to equipment labeled as "fragile", so the event "person 1 $moves\ close\ to$ an equipment" is triggered. The first event of scenario "Period near fragile equipment" is instanciated.

Further at t_3 Person 1 is still $close$ to the machine, so the negative event of scenario "Period near fragile equipment" is instanciated. Secondly, the fact that Person 1 stops triggers the event "Person 1 $stops$". The three events of the scenario "Period near fragile equipment" are recognized and an alarm is sent to the security operator. The complete recognition of this scenario triggers the specific loopback event: $presence\_near\_fragile$" equipment. This specific event matches the first event of the scenario "Repeated Presence period near fragile equipment". Then at t_4 (see Fig. 14) an

Scenario

Name = "Repeated period near fragile equipment",
Events =
(t_1, presence_period_near_fragile, e_1, p_1),
($t_2, moves\ close\ to\ (p_1 : Person,\ e_1 : Equipment)$),
not($t_3,\ moves\ away\ from\ (p_1 : Person,\ e_1 : Equipment)$),
($t_4,\ stop(p_1 : Person)$)
Constraints =
$t_1 \leq t_2,\ t_2 \leq t_3,\ t_2 \leq t_4,$
$t_3 \leq t_2$ + normal_presence_time(e_1),
Conditions = function(e_1, "fragile"),
Success = alarm("Vandalism on ",e_1)

Figure 12. AVS-PV scenario model: "Repeated Presence period near fragile equipment"

Figure 13. Left: Metro entrance in Nuremberg at t_2. Right: an event "person 1 $moves\ close\ to$ an equipment" is detected.

Figure 14. Left: Metro entrance in Nuremberg at t_4. Right: an event 'person 1 $moves\ away\ from$ an equipment" is detected

Figure 15. Left: Metro entrance in Nuremberg at t_5. Right: a "Vandalism' 'alarm is sent to the security operator.

other event is detected because the Person 1 moves away in the direction of the corridor to check is anybody is arriving.

At t_5 (see Fig. 15), the event "Person 1 $moves\ close\ to$ equipment" is triggered. This equipment is the same equipment that the one at t_2. The scenario "Repeated Presence period near fragile equipment" is now totally recognized. An alarm is sent to the security operator.

The results of these applications were considered very satifactory by the metro operators. The richness of the formalism for scenario description allows to specify a set of constraints (temporal as well as atemporal ones) which reduce false alarms. The formalism we have proposed for scenario description has enabled us to represent the expertise for these applications. The knowledge modeling is still difficult. The main reason is that we need to manage the passage from vague security concepts (such as "abnormal behavior") to rigorous scenario models. These results have been processed off-line on a Sun Ultra10 workstation. The

computing time per image is between 220ms and 530ms for the complete chain (including perception and interpretation). Among the 25 images digitized per second, 5 images (one per 200ms) are processed.

7. Conclusion

In this paper we have shown that high-level video understanding can be performed based on images taken from a single static camera and with simple perception methods working almost in real-time. This has been possible by using two sets of a priori information: first, contextual information describing the $3D$ geometry of the observed scene and semantic information on the static objects and interesting areas, second, general knowledge of predefined scenarios valid for an application domain. We have proposed a formalism to represent these two types of a priori information and explained how to use them for video understanding. We have also proposed a formalism for event recognition based on object state models. This formalism is independent of a particular application domain and enables the passage between the perception data and the scenario models. The current video understanding framework we propose has shown several limitations. One type of problems is the imprecision and uncertainty in the detection and location of mobile objects; most of these low-level detection errors are due either to reflections, shadows or occlusions. A solution to cope with these problems is to relax our second hypothesis and not to restrict ourselves to the use of a single camera. Another more general problem is that as every vision system, this framework needs, for each perception method and for each interpretation method, to set the values of numerical parameters in a configuration phase. One solution to solve this problem is to use learning techniques to find the best parameter values for an application.

References

[1] E. André, G. Herzog, and T. Rist. On the simultaneous interpretation of real world image sequences and their natural language description: The system soccer. In *8th European Conference of Artificial Intelligence*, pages 449 – 454, Munich, 1988.

[2] D. Ayers and M. Shah. Monitoring human behavior in an office environment. In *Computer Society Workshop on Interpretation of Visual Motion*, 1998.

[3] H. Buxton and S. Gong. Visual surveillance in dynamic and uncertain world. *Artificial Intelligence Journal*, 78:431 – 459, 1995.

[4] C. Castel, L. Chaudron, and C. Tessier. What is going on ? a high level interpretation of sequences of images. In *4th European Conference on Computer Vision, Workshop on Conceptual Descriptions from Images*, Cambridge UK, April 1996.

[5] N. Chleq, F. Bremond, and M. Thonnat. *Advanced Video-Based Surveillance Systems*, chapter Image Understanding for Prevention of Vandalism in Metro Station, pages 106 – 117. Kluwer Academic Publishers, 1999.

[6] N. Chleq and M. Thonnat. Realtime image sequence interpretation for videosurveillance. In IEEE, editor, *International Conference on Image Processing*, pages 801 – 804, Lausanne, Switzerland, 1996.

[7] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. In *Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 1999.

[8] S. Dettmer, A. Seetharamaiah, L. Wang, and M. Shah. Model-based approch for recognizing human activities from video sequences. In *Workshop on Motion of Non-Rigid and Articulated Objects*, June 1998.

[9] C. Dousson, P. Gaborit and M. Ghallab. Situation recognition:representation and algorithms. In *Proc. 13th IJCAI*, Chambery, p.166-172, 1993.

[10] G. Herzog. Utilizing interval-based event representation for incrementatal high-level scene analysis. In *4th International Workshop on Semantics of Time, Space, and Movement and Spatio-Temporal Reasonning*, Chateau de Bonas, France, 1992.

[11] S. Intille and A. Bobick. Closed world tracking. In *5th International Conference on Computer Vision*, Cambridge, 1995.

[12] S. Intille and A. F. Bobick. Visual recognition of multi-agent action using binary temporal relations. In *Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 1999.

[13] Y. Ivanov, C. Stauffer, A. Bobick, and W. Grimson. Video surveillance of interactions. In *2nd International Workshop on Visual Surveillance*, pages 82 – 89, Fort Collins, Colorado, June 1999.

[14] G. Medioni, I. Cohen, F. Brémond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. In *DARPA Image Understanding Workshop*, Monterey, November 1998.

[15] N. Rota, R. Stahr and M. Thonnat. .Tracking for Visual Surveillance in VSIS. In *PETS2000*, Grenoble, March 2000.

[16] C. Tessier. Reconnaissance de scènes dynamiques à partir de données issues de capteurs: le projet perception. Technical report, Onera-Cert, 2 avenue Edouard-Belin BP 4025 31055 Toulouse Cedex France, Août 1997.

[17] M. Thonnat and N. Rota. Image understanding for visual surveillance applications. In *Third International Workshop on Cooperative Distributed Vision*, Kyoto, Japan, November 1999.