Automatic visual recognition for metro surveillance

F. Cupillard, M. Thonnat, F. Brémond
Orion Research Group, INRIA, Sophia Antipolis, France

Abstract

We propose in this paper an approach for recognizing either isolated individual, group of people or crowd behaviors in the context of visual surveillance of metro scenes using multiple cameras. In this context, a behavior recognition module relies on a vision module composed of three tasks: (a) motion detection and frame to frame tracking, (b) multiple cameras combination and (c) long term tracking of individuals, groups of people and crowd evolving in the scene. For each tracked actor, the behavior recognition module performs three levels of reasoning: states, events and scenarios. We have also defined a general framework to easily combine and tune various recognition methods (e.g. automaton, Bayesian network or AND/OR tree) dedicated to the analysis of specific situations (e.g. mono/multi actors activities, numerical/symbolic actions or temporal scenarios). Validation results on different methods used to recognize specific behaviors are described.

Keywords

Visual surveillance, behavior recognition, real-time.

1 Introduction

In this article, we present a method for recognising specific people behaviors such as fighting or vandalism occurring in a cluttered scene (typically a metro scene) viewed by several cameras. The development of visual surveillance systems, as proposed by Hongeng [1], Pentland [2] and Xiang [3], presents several difficulties and one of the most challenging is behavior analysis, since it requires the inference of a semantic description of the features (moving regions, trajectories,...) extracted from the video stream. Our ambitious goal is to recognize in real time behaviors involving either isolated individuals, groups of people or crowd from real world video streams coming from metro stations. This work is performed in the framework of the European project ADVISOR (http://www-sop.inria.fr/orion/ADVISOR).To reach this goal, we developed a system which takes as input video streams coming from cameras and generates annotation about the activities recognized in the video streams. The paper is organised as follows: in the first part, we present briefly the global system and its vision module. Then, we detail the behavior recognition process illustrated through three behavior recognition examples: "Fighting", "Blocking" and "Fraud" behaviors.

2 Overall System Overview

The video interpretation system is based on the co-operation of a vision and a behavior recognition module as shown on Figure 1.

The vision module is composed of three tasks. First a motion detector and a frame to frame tracker generates a graph of *mobile objects* for each calibrated camera. Second, a combination mechanism is performed to combine the graphs computed for each camera into a global one. Third, this global graph is used for long term tracking of individuals, groups of people and crowd evolving in the scene (typically on hundreds of frames). For each tracked actor, the behavior recognition module performs three levels of reasoning: states, events and scenarios. On top of that, we use 3D scene models, one for each camera, as a priori contextual knowledge of the observed scene. We define in a scene model the 3D positions and dimensions of the static scene objects (e.g. a bench, a ticket vending machine) and the zones of interest (e.g. an entrance zone). Semantic attributes (e.g. fragile) can be associated to the objects or zones of interest to be used in the behavior recognition process.

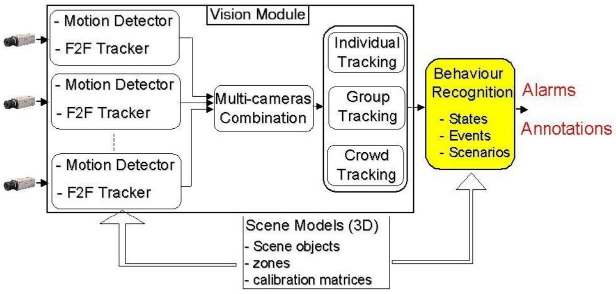


Figure 1. *Video interpretation system.*

2.1 Motion Detector and Frame to Frame Tracking

The goal of the Motion Detector is to detect for each frame the moving regions in the scene and classify them into a list of *mobile objects* with labels corresponding to their type based on their 3D size, such as *PERSON*. This task can be divided into three sub-tasks: detection of *mobile objects*, extraction of features, classification of *mobile objects*. A list of *mobile objects* is obtained at each frame. Each *mobile object* is described by 3D numerical parameters (center of gravity, position ,height, width,...) and by a semantic class (*PERSON, OCCLUDED PERSON, GROUP, CROWD, METRO TRAIN, SCENE OBJECT, NOISE or UNKNOWN*).

The goal of the frame to frame tracker (F2F Tracker) is to link from frame to frame the list of *mobile objects* computed by the motion detector. The output of the frame to frame tracker is a graph of *mobile objects*. This graph provides all the possible trajectories that a *mobile object* may have. The link between a new *mobile object* and an old one is computed depending on three criteria: the similitude between their semantic classes, their 2D (in the image) and their 3D (in the real world) distance.

2.2 Multiple cameras Combination

In order to take advantage of all the calibrated cameras viewing the same scene (cameras with overlapping field of views), we combine all the graphs of *mobile objects* computed by the F2F Tracker for each camera into a global one that we called the **Combined** Graph (see [7] for more details). As a result, the features (the 3D positions and the dimensions) of the *mobile objects* computed in the Combined Graph give a better estimation of the positions and the dimensions of the real persons evolving in the scene.

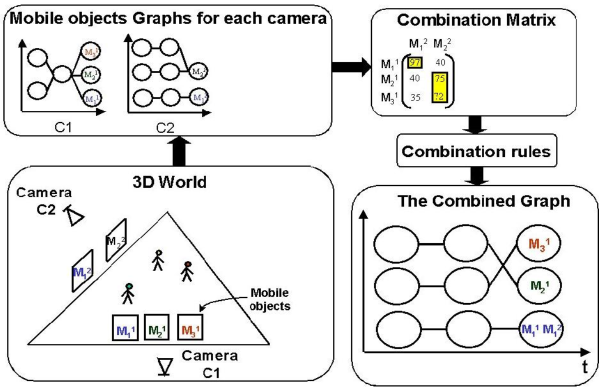


Figure 2. *This figure illustrates the multiple cameras combination process. Three persons are evolving in the scene. Camera C1 detects three mobile objects whereas camera C2 detects only two mobile objects. The combination matrix enables to determine (a) a high correspondence between the mobile object M_1^1 of C1 and the mobile object M_1^2 of C2; these two mobile objects are fused together in the combined graph, and (b) an ambiguous correspondence between the two mobile objects M_2^1 and M_3^1 of C1 and the mobile object M_2^2 of C2; the two mobile objects M_2^1 and M_3^1 detected by C1 are selected in the combined graph.*

To compute the global graph, we combine at each frame the new *mobile objects* detected for 2 cameras using a combination matrix and a set of rules (see illustration on Figure 2).

The combination matrix gives the correspondences between the *mobile objects* detected for two cameras by using a 3D position and a 3D size criteria. In the case of none ambiguities between the *mobile objects* detected by the two cameras, we fuse the *mobile objects* by making an average on their 3D features. In case of ambiguities, a set of rules is used to either select or eliminate the *mobile object* detected by one of the two cameras.

2.3 Individual, Group of people and Crowd Long Term Tracking

The goal here is to follow on a long period of time either Individuals, Groups of people or Crowd to allow the scenarios involving these three different types of actors to be recognized. For example, when we want to detect a group of people (at least two persons) which is blocking an exit zone, we prefer reasoning with the Group Tracker because it provides a more accurate 3D location of the group of people in the scene.

The Individual Tracker tracks each person individually whereas the Group Tracker tracks globally all the persons belonging to the same group. Both trackers perform a temporal analysis of the Combined Graph. The Individual Tracker computes and selects the trajectories of *mobile objects* which can correspond to a real person thanks to an explicit model of person trajectory. In a similar way, the Group Tracker computes and selects the trajectories of *mobile objects* which can correspond to the persons inside a real group thanks to an explicit model of the trajectories of people inside a group.

Individual and Group Trackers are running in parallel. When the density (computed over a temporal window) of detected *mobile objects* becomes too high (typically if the *mobile objects* overlap more than 2/3 of the image), we stop these two trackers because in such a situation, they cannot give reliable results. At this point, we trigger the Crowd Tracker which is in fact the Group Tracker with an extended model of the trajectories of people inside a group allowing a large density of detected people belonging to the same group that by this way defines a crowd.

3 Behavior Recognition Process

The goal of this task is to recognize specific behaviors occurring in a metro scene. A main problem in behavior recognition is the ability to define and reuse methods to recognize specific behaviors, knowing that the perception of behaviors is strongly dependent on the site, the camera view point and the individuals involved in the behaviors. Our approach consists in defining a formalism allowing us to write and easily reuse all methods needed for the recognition of behaviors. This formalism is based on three main ideas. First the formalism should be flexible enough to allow various types of operators to be defined (e.g. a temporal filter or an automaton). Second, all the needed knowledge for an operator should be explained within the operator so that it can be easily reused. Finally, the description of the operators should be declarative in order to build an extensible library of operators.

3.1 Behavior representation

We call an actor of a behavior any scene object involved in the behavior, including static objects (equipment, zones of interest…), individuals, groups of people or crowd. The entities needed to recognize behaviors correspond to different types of concepts which are:

1. **The basic properties**: a characteristic of an actor such as its trajectory or its speed.

2. **The states**: a state describes a situation characterising one or several actors defined at time *t* (e.g. a group is agitated) or a stable situation defined over a time interval. For the state: "an individual stays close to the ticket vending machine", two actors are involved: an individual and a piece of equipment.

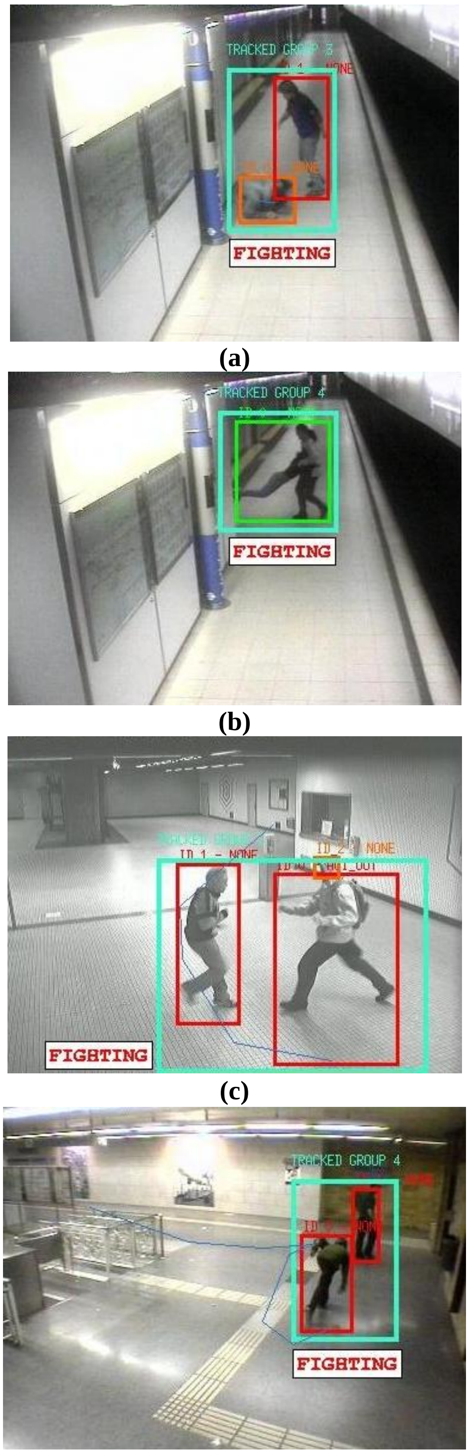
3. **The events**: an event is a change of states at two consecutive times (e.g. a group enters a zone of interest).

4. **The scenarios**: a scenario is a combination of states, events or sub scenarios. Behaviors are specific scenarios (dependent on the application) defined by the users. For example, to monitor metro stations, end-users have defined 5 targeted behaviors: "Fraud", "Fighting" "Blocking", "Vandalism" and "Overcrowding".

To compute all the needed entities for the recognition of behaviors, we use a generic framework based on the definition of **Operators** which are composed of four attributes:

Operator name: indicates the entity to be computed such as the state "an Individual is walking" or "the trajectory is straight".

Operator input: gives a description of input data. There are two types of input data: basic properties characterising an actor and sub entities computed by other Operators.



(a)

(b)

(c)

(d)

Figure 3. *This figure illustrates four methods combined by an AND/OR tree to recognize the behavior "Fighting". Each image illustrates a configuration where one method is more appropriate to recognize the behavior: (a) lying person on the floor surrounded by people, (b) significant variation of the group width, (c) quick separation of people inside a group and (d) significant variation of the group trajectory.*

Operator body: contains a set of competitive methods to compute the entity. All these methods are able to compute this entity but they are specialised depending on different configurations. For example, to compute the scenario "fighting", there are 4 methods (as shown on Figure 3). For example, one method computes the evolution of the lateral distance between people inside a group. A second one detects if someone, surrounded by people, has fallen on the floor.

Operator output: contains the result of the entity computation accessible by all the other Operators. This result corresponds to the value of the entity at the current time.

This generic framework based on the definition of Operators gives two advantages: It first enables us to test a set of methods to compute an entity, independently of other entities. So we can locally modify the system (the methods to compute an entity) while keeping it globally consistent (without modifying the meaning of the entity). Second, the network of Operators to recognize one scenario is organised as a hierarchy. The bottom of the hierarchy is composed of states and the top corresponds to the scenario to be recognized. Several intermediate levels, composed of state(s) or event(s) can be defined.

3.2 Behavior recognition

We have defined four types of methods depending on the type of entities:

Basic properties methods: we use dedicated routines to compute properties characterising actors such as trajectory, speed and direction. For example, we use a polygonal approximation to compute the trajectory of an individual or a group of people.

State methods: we use numerical methods which
include the computation of: (a) 3D distance for states dealing with spatial relations (e.g. "an individual is close to the ticket vending machine"), (b) the evolution of temporal features for states dealing with temporal relations (e.g. "the size of a group of people is constant") and (c) the speed for states dealing with spatio-temporal relations (e.g. "an individual is walking") and (d) the combination of sub states computed by other operators.
The output of these numerical methods is then classified to obtain a symbolic value.

Event methods: we compare the status of states at two consecutive instants. The output of an event method is boolean: the event is either detected or not detected. For example, the event "a group of people enters a zone of interest" is detected when the state "a group of people is inside a zone of interest" changes from false to true.

Scenario methods: for simple scenarios (composed of only 1 state), we verify that a state has been detected during a predefined time period using a temporal filter. For sequential scenarios (composed of a sequence of states), we use finite state automatons. An automaton state corresponds to a state and a transition to an event. An automaton state also corresponds to an intermediate stage before the complete recognition of the scenario. We have used an automaton to recognize the scenarios "Blocking" and "Fraud" as described on Figure 4 and 5.
For composed scenarios defining a single unit of movement composed of sub scenarios, we use Bayesian networks as proposed by Hongeng [4] or AND/OR trees of sub scenarios as illustrated on Figure 6. A description of Bayesian networks for scenario recognition can be found in [6]. We have defined one Bayesian network to recognize the "violence" behavior composed of 2 sub scenarios: "internal violence" (e.g. erratic motion of people inside a group) and "external violence" (e.g. quick evolution of the trajectory of the group).

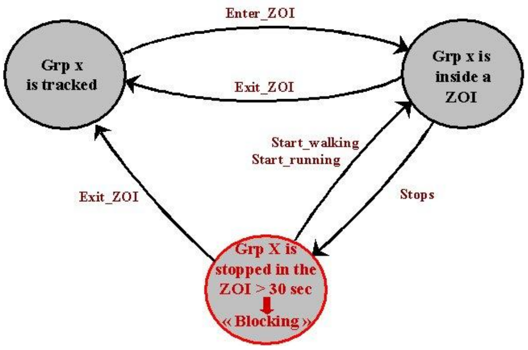


Figure 4. *To check whether a group of people is blocking a zone of interest (ZOI), we have defined an automaton with three states: (a) a group is tracked, (b) the group is inside the ZOI and (c) the group has stopped inside the ZOI for at least 30 seconds.*

Both of these methods need a learning stage to learn the parameters of the network using ground truth (videos annotated by operators). Bayesian networks are optimal given ground truth but the AND/OR trees are easier to tune and to adapt to new scenes.

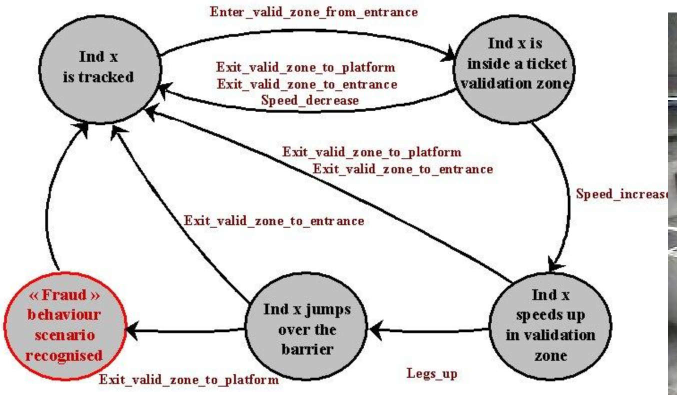

Enter_valid_zone_from_entrance
Ind x
is tracked
Exit_valid_zone_to_platform
Exit_valid_zone_to_entrance
Speed_decrease
Ind x is
inside a ticket
validation zone
Exit_valid_zone_to_platform
Exit_valid_zone_to_entrance
Speed_increase
Exit_valid_zone_to_entrance
« Fraud »
behaviour
scenario
recognised
Ind x jumps
over the
barrier
Ind x
speeds up
in validation
zone
Legs_up
Exit_valid_zone_to_platform

Figure 5. *To check whether an individual is jumping over the barrier without validating his ticket, we have defined an automaton with five states: (a) an individual is tracked, (b) the individual is at the beginning of the validation zone, (c) the individual has a high speed, (d) the individual is over the barrier with legs up and (e) the individual is at the end of the validation zone.*

For scenarios with multiple actors involved in complex temporal relationships, we use a network of temporal variables representing sub scenarios and we backtrack temporal constraints among the already recognized sub scenarios as proposed by Van Vu [5].

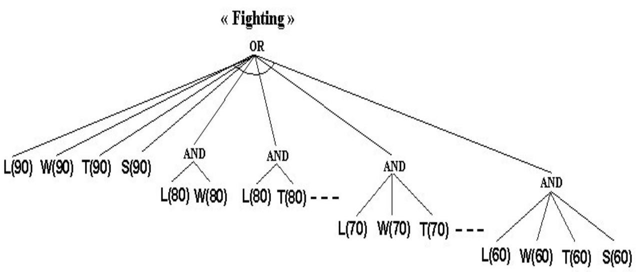


Figure 6. *To recognize whether a group of people is fighting, we have defined an AND/OR tree composed of four basic scenarios: (L) lying person on the floor surrounded by people, (W) significant variation of the group width, (S) quick separation of people inside the group and (T) significant variation of the group trajectory. Given these four basic scenarios we were able to build an OR node with all combinations (corresponding to 15 sub scenarios) of the basic scenarios. These combinations correspond to AND nodes with one up to four basic scenarios. The more basic scenarios there are in AND nodes, the less strict is the recognition threshold of each basic scenario. For example, when there is only one basic scenario (e.g. L(90)), the threshold is 90 and when there are four basic scenarios, the threshold is 60. To parameterise these thresholds, we have performed a learning stage consisting in a statistical analysis of the recognition of each basic scenario.*

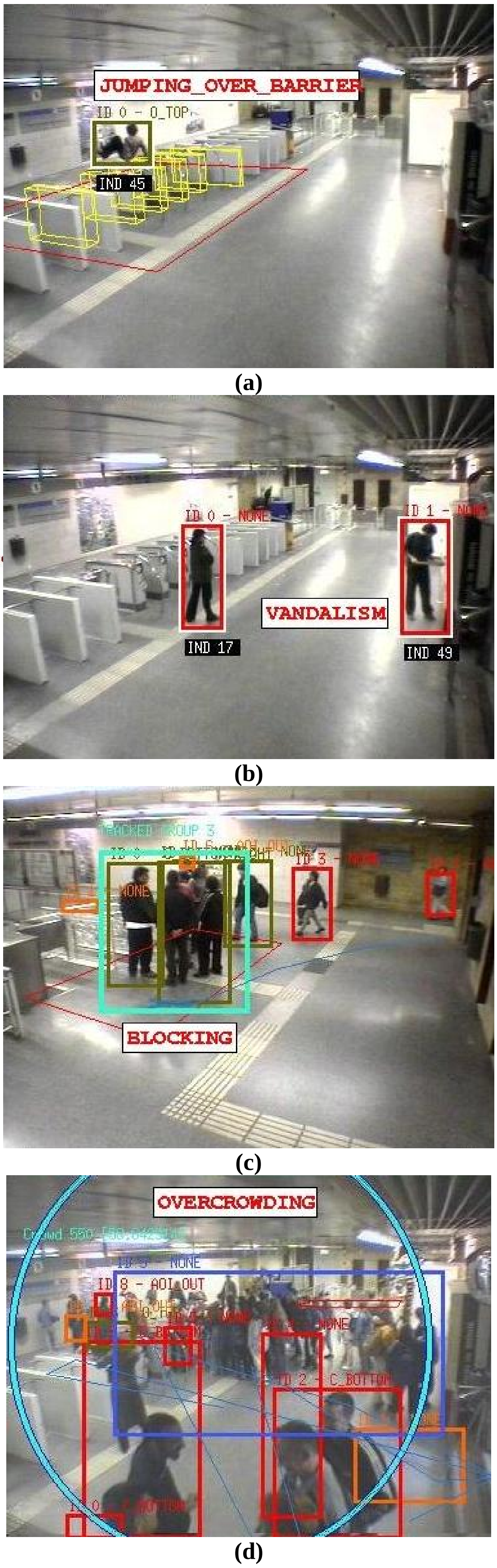


Figure 7. *This figure illustrates four behaviors selected by end users and recognized by the video interpretation system: (a) "Fraud" recognized by an automaton, (b) "Vandalism" recognized by a temporal constraint network, (c) "Blocking" recognized by an automaton and (d) "Overcrowding" recognized by an AND/OR tree.*

3.3 Behavior recognition results

The behavior recognition module is running on a PC Linux and is processing four tracking outputs corresponding to four cameras with a rate of 5 images per second. We have tested the whole video interpretation system (including motion detection, tracking and behavior recognition) on videos coming from ten cameras of Barcelona and Brussels metros. We correctly recognized the scenario "Fraud" 6/6 (6 times out of 6) (Figure 7.a), the scenario "Vandalism" 4/4 (Figure 7.b), the scenario "Fighting" 20/24 (Figure 3), the scenario "blocking" 13/13 (Figure 7.c) and the scenario "overcrowding" 2/2 (Figure 7.d). We also tested the system over long sequences (10 hours) to check the robustness over false alarms. For each behavior, the rate of false alarm is: 2 for "Fraud", 0 for "vandalism", 4 for "fighting", 1 for "blocking" and 0 for "overcrowding".

Moreover, in the framework of the European project ADVISOR, the video interpretation system has been ported on Windows and installed at Barcelona metro in March 2003 to be evaluated and validated. This evaluation has been done by Barcelona and Brussels videosurveillance metro operators during one week at the Sagrada Familia metro station. Together with this evaluation, a demonstration has been performed to various guests, including the European Commission, project Reviewers and representative of Brussels and Barcelona Metro to validate the system. The evaluation and the demonstration were conducted using both live and recorded videos: four channel in parallel composed of three recorded sequences and one live input stream from the main hall of the station. The recorded sequences enabled to test the system with rare scenarios of interest (e.g. *fighting*, *jumping over the barrier*, *vandalism)* whereas the live camera allowed to evaluate the system against scenarios which often happen (e.g. *overcrowding*) and which occurred during the demonstration and also to evaluate the system against false alarms. In total, out of 21 *fighting* incidents in all the recorded sequences, 20 alarms were correctly generated, giving a very good detection rate of 95%. Out of nine *blocking* incidents, seven alarms were generated, giving a detection rate of 78%. Out of 42 instances of *jumping over the barrier*, including repeated incidents, the behavior was detected 37 times, giving a success rate of 88%. The two sequences of *vandalism* were always detected over the six instances of *vandalism*, giving a perfect detection rate of 100%. Finally, the *overcrowding* incidents were also consistently detected in the live camera, with some 28 separate events being well detected.

In conclusion, the ADVISOR demonstration has been evaluated very positively by end-users and European Committee. The algorithm responded very successfully to the input data, with high detection rates, less than 5% of false alarms and with all the reports being above approximately 70% accurate.

4 Conclusion and Future Work

In this article, we have described a video interpretation system able to automatically recognize high level of human behaviors involving individuals, groups of people and crowd.

Different methods have been defined to compute specific types of behaviors under different configurations. All these methods have been integrated in a coherent framework enabling to modify locally and easily a given method. The system has been fully tested off-line and has been evaluated, demonstrated and successfully validated in live condition during one week at the Barcelona metro in March 2003. The next step consists in designing the video interpretation system to be operational (able to cope with any unpredicted real world event) and working on a large scale. For that, we need to design a platform able to be configured dynamically and automatically.

References

1. Hongeng, S.; Brémond, F.; Nevatia, R. (2000). Bayesian framework for video surveillance application. *Proc. of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, Sept. 2000.
2. Pentland, A.; Liu, A. (1999). Modeling and prediction of human behavior. In : *Neural Computation*, pp. 229-242.
3. Xiang, T.; Gong, S.; Parkinson, D. (2003). On the structure of dynamic Bayesian networks for complex scene modelling. *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS).* Nice, France, Oct. 2003.
4. Hongeng, S.; Nevatia, R. (2001). Multi-agent event recognition. *International Conference on Computer Vision (ICCV2001)*, Vancouver, B.C., Canada, 2001/07/12.
5. Van Vu, T.; Brémond, F.; Thonnat, M. (2003). Automatic video interpretation: A recognition algorithm for temporal scenarios based on pre-compiled scenario models. *ICVS2003.*
6. Moenne-Loccoz, N.; Brémond, F.; Thonnat, M. (2003). Recurrent Bayesian network for the recognition of human behaviors video, *ICVS2003*.
7. Cupillard, F.; Brémond, F.; Thonnat, M. (2002). Group behavior recognition with multiple cameras. *IEEE Workshop on Applications of Computer Vision*, Orlando, Dec. 2002.