Evaluation and Knowledge Representation Formalisms to Improve Video Understanding

Benoît Georis Magali Mazière François Brémond
Monique Thonnat
INRIA-ORION team
2004 route des Lucioles
BP 93, 06902 Sophia-Antipolis, France
firstname.lastname@sophia.inria.fr

Abstract

This article presents a methodology to build efficient real-time semantic video understanding systems addressing real world problems. In our case, semantic video understanding consists in the recognition of predefined scenario models in a given application domain starting from a pixel analysis up to a symbolic description of what is happening in the scene viewed by cameras. This methodology proposes to use evaluation to acquire knowledge of programs and to represent this knowledge with appropriate formalisms. First, to obtain efficiency, a formalism enables to model video processing programs and their associated parameter adaptation rules. These rules are written by experts after performing a technical evaluation. Second, a scenario formalism enables experts to model their needs and to easily refine their scenario models to adapt them to real-life situations. This refinement is performed with an end-user evaluation. This second part ensures that systems match end-user expectations. Results are reported for scenario recognition performances on real video sequences taken from a bank agency monitoring application.

1. Introduction

Many video understanding systems have already been developed in the computer vision community. Haritaoglu et al. [6] use shape analysis and tracking to locate people and their parts (e.g., head, feet) in image sequences. Oliver et al. [9] use Bayesian analysis to identify human interactions using trajectories obtained from a monocular image. Johnson and Hogg [7] have defined an efficient people tracker based on B-spline corresponding to people shape models. Nevertheless, few video understanding systems have been successfully applied to real world applications due to a large variety of video understanding issues. First, typical image processing problems come from shadows, illumination changes, over-segmentations or mis-detections. Second, the tracking process remains a major issue since the loss of a tracked object prevents the analysis of its behaviour. In addition, few systems provide a true semantic video understanding. [12] is part of the few examples able to perform complex reasoning (i.e., spatio-temporal reasoning) and to understand people interactions in real world applications. Finally, systems usually perform well on a small set of video sequences or in a well-constrained environment but results worsen in real conditions. Despite these facts, there is an increasing number of installed video surveillance systems being run 24 hours a day in varying conditions. Therefore, there is a strong need for highly reliable and adaptive systems with more and more reasoning capabilities. The recent creation of PETS (Performance Evaluation of Tracking and Surveillance) workshops [5] shows the concern of the vision community to address this issue. It enforces the idea that evaluation techniques are needed to assess the reliability of algorithms. On the system architecture level, recent works address software engineering issues. In [3], [10] and [1] authors propose generic and modular architectures for video surveillance systems. For example, experts can easily add and test new algorithms through a plug-and-play property and several interactive tools which display results and assess qualitative performances. Nevertheless, such architectures lack a mean to formalize the knowledge acquired by experts and do not have high-level reasoning capabilities. In [8], the architecture is endowed with a control strategy through a rule-based supervisor expressed in Clips or Prolog languages. This dynamic configuration ability is demonstrated on a smart office application. This approach does not use intensively 3D knowledge or contextual information though and there is no formalism to represent the knowledge. To obtain a robust system, we need

of course a modular software architecture but above all a methodology to structure, represent and efficiently use all the knowledge which is needed to obtain efficient systems. The proposed methodology combines knowledge representation formalisms with an evaluation framework to reach this goal. First, a formalism which relies on the concept of video processing operators enables to represent the knowledge of video processing programs such as parameter initialization rules or evaluation rules. These rules are formalized by a video processing expert and given to a system as a priori knowledge. A technical evaluation helps experts to acquire this knowledge. Second, a scenario formalism enables end-users to express their needs. This formalism also enables to easily refine these scenarios during an end-user evaluation. This evaluation is an interactive process which is intended to acquire knowledge of the application domain and which ensures that recognized scenarios correspond to end-user expectations. The paper is organized as follows. Section 2 describes the proposed methodology to build adaptive and efficient video understanding systems. Section 3 describes the formalism used to represent video processing programs and the associated technical evaluation. Section 4 presents the scenario representation formalism and explains the end-user evaluation which enables a scenario model refinement and which ensures that recognized scenarios correspond to real needs. Finally section 5 concludes and indicates future work.

2. Video understanding

2.1. Video understanding modelling

Before presenting the methodology, we describe a model of a typical video understanding process which is illustrated in Fig. 1. This model is useful to guide experts in their formalization of knowledge of video processing programs (e.g., how to group different techniques under the same abstract functionality). First, colour input images are digitized with a variable frame rate provided by one or several cameras. For each camera, a procedure detects moving regions by subtracting the current image from a reference image and classifies them according to a semantic class (e.g., *person*, *vehicle*) of mobile objects. Then, a F2F (Frame to Frame) tracking process links mobile objects over time and generates a graph of mobile objects for each camera. Nodes of this graph represent mobile objects while edges are temporal links over time. The various sequences of edges in this graph represent the various possible trajectories a mobile object may have. Second, a multi-camera procedure combines the graphs coming from the different cameras with overlapped field of view in order to obtain a unique 3D representation of mobile objects. Then, this global graph is processed by a long term tracking process which computes a set of paths representing the possible trajectories on a large time window, typically several seconds. Its purpose is to robustly track actors evolving in the scene by comparing the evolution of the various paths which can be followed and choosing the best path to update the mobile object track. Finally, scenario recognition is performed for all tracked actors. So, the output is a semantic description of the recognized scenarios. In addition, 3D scene models are used as a priori contextual knowledge of the observed scene. For instance, a scene model contains 3D positions and dimensions of static scene objects (e.g., a ticket vending machine, a chair) and zones of interest (e.g., an entrance zone). Semantic attributes (e.g., fragile) are associated to objects or zones to be used in the scenario recognition process. Application domain knowledge (scenario model library) and video processing knowledge (video processing operators) are also fed into the system. The challenge is how to organize the video processing programs and optimize this knowledge to obtain an efficient processing.

2.2. Formalism utilization methodology

Despite the large amount of existing libraries of video processing programs and sophisticated architectures, it is difficult to obtain a deep knowledge of these programs (e.g., which technique is more suitable on which data under which environmental conditions). These systems lack adaptivity to changing environmental conditions and reusability from one application to another one. We want to insist on the fact that we have a large amount of knowledge available [2] to help us to build efficient systems but we need to structure, capitalize and use this knowledge efficiently. In other words, we would like to have the same good software engineering properties at the knowledge engineering level. The proposed methodology takes place within the paradigm of knowledge-based techniques. We propose to use knowledge representation formalisms in order to obtain three properties:

- Isolated knowledge: knowledge is extracted and separated away from the code and represented in a knowledge base with formalisms. This way, knowledge is both human and machine readable. We can thus progressively capitalize human experience and expertise. This separation between code and knowledge also allows to better control the tasks at hand.

- Modular knowledge: knowledge is structured in small parts corresponding to clearly identified video processing functionalities. For instance, a piece of knowledge can be dedicated to adapt a program parameter to an environmental condition, thus contributing to the adaptivity property.

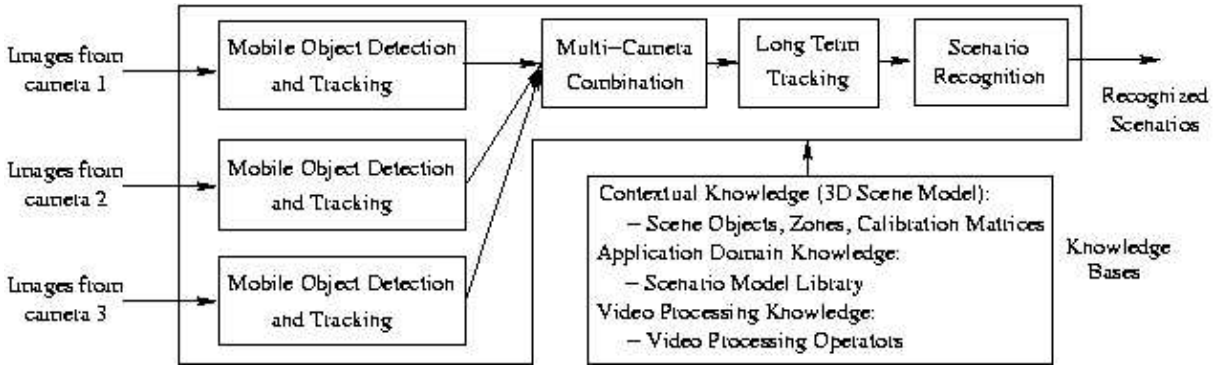

Figure 1. A typical video understanding process composed of four main tasks.

- Upgradable knowledge: once the expert discover a new piece of knowledge (e.g., how to tune a segmentation parameter when passing from indoor to outdoor scenes), he/she can add it easily while keeping a good knowledge structure. Knowledge is easily maintained and/or extended.

Moreover, based on our experience in dealing with complex problems such as video understanding, knowledge acquisition is a challenging objective. Indeed, the variability of input data is tremendous. It is thus difficult to understand the impact of each variation (e.g., in the scene configuration) on program performances. We must be sure that adding a piece of knowledge at a given level of processing will not cause new problems elsewhere in the process, due to dependencies. That is why we propose to combine the use of knowledge representation formalisms with an automatic evaluation framework. First, the evaluation can be run on a large set of test video sequences. An expert can thus thoroughly understand a problem and verify that the proposed piece of knowledge effectively solve the problem. Second, the evaluation is performed automatically for each functionality or subfunctionality in the system, thus providing the knowledge modularity property. This automatic evaluation framework is a key element to build an operational knowledge base. We now present the operator formalism which enables to define properly the video understanding process in order to perform a pertinent evaluation and to optimize programs.

3. Video processing knowledge representation

3.1. Operator formalism

The proposed formalism to model video processing programs is inspired by a formalism previously introduced for image processing [11]. The main concept is the notion of operator which represents a video processing program as illustrated in Fig. 2. An operator can either be primitive or composite (hierarchy of operators). It contains several information to guide the reasoning process: functionality, data types, pre and post-conditions, tunable parameters, and a list of rules. There are four types of rules: parameter initialization, operator selection, result evaluation, repair rules. An initialization rule ensures a correct value for an operator parameter when this operator is selected for execution. For instance, a 3D distance threshold for a tracking operator can be set knowing the usual walking speed of a human being. An evaluation rule assesses whether obtained results are satisfactory or not. A repair stage can be triggered in two ways: a bad performance evaluation or a detection of an environmental change. Repair may consist in adjusting dynamically a parameter or in selecting a new operator. The repair can be done at the primitive or composite operator level. The scope of repair is either global (valid for all data and for the future until the next change) or local (valid for a subset of data and the current frame only, i.e., data are reprocessed). For instance, given an environmental change, a repair rule may decide to switch from an individual tracker operator to a crowd tracker operator when people get out of a train in a subway. A bad performance evaluation may dynamically trigger a decrease of a segmentation operator threshold in order to detect mobile objects with low contrast with respect to background. All these rules are used by a reasoning engine which controls all programs (i.e., the control is external to programs). This is the focus of another paper [11]. Nevertheless, two main problems arise from knowledge-based approaches: 1) knowledge acquisition is tedious, 2) resulting systems are often complex and with poor performances. Therefore, we focus now on showing that a technical evaluation may help to ease knowledge acquisition.

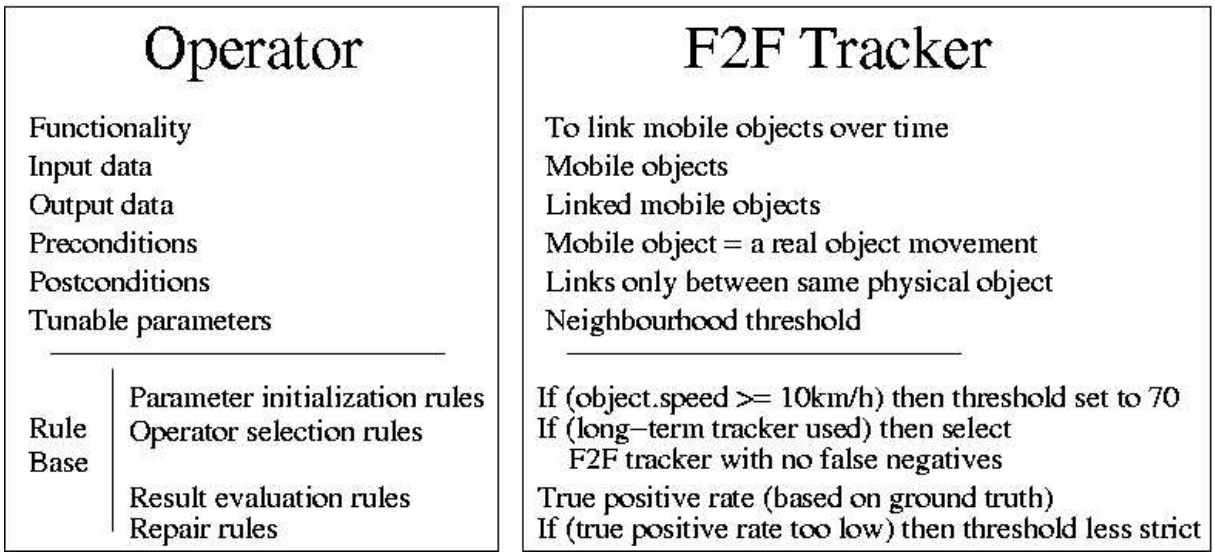

Figure 2. Operator general description (on the left) and an example for the F2F tracking process (on the right).

Figure 3. Example of images coming from a bank monitoring application.

3.2. Technical evaluation

In order to obtain a pertinent evaluation, we must study exhaustively all testing conditions. The goal is to classify video sequences according to several difficulty criteria (e.g., level of clutter, amount of illumination changes) and to select a representative set of test video sequences containing these difficulties ranging from easy to hard. This method enables to establish relations between a difficulty and its impact on performances. An example of processed images is presented in Fig. 3. The most difficult video sequences selected for the technical evaluation in a bank monitoring application were containing 4 people crossing each other (2 bank employees, 1 robber and 1 customer) and 3 contextual objects (movable chair, counter and safe gate) during 400 frames. These videos were interesting since they highlighted a chair displacement problem for the mobile object detection operator, frequent people crossings for the tracking operator and complex composite scenarios for the scenario recognition operator. In addition, the system has been evaluated on hours of live and recorded video sequences without ground truth. Supervised evaluation is the most accurate evaluation we can obtain and thus the main way to acquire knowledge of a video processing operator behaviour (which performances in which conditions). There are two supervised evaluation types. First, the evaluation

is performed using directly user-interaction. For example, the user can globally indicate the number of people in the scene, which thus gives a direct feedback to the long-term tracking operator. Secondly, the evaluation is performed using ground truth. In this case, ground truth data must be defined as objectively as possible as discussed in [4]. For each mobile object, stored attributes are the 2D width and height, the 2D position and an identifier. Once ground truth data have been acquired, we are able to perform an automatic evaluation on a large set of test video sequences.

3.3. results

We report here the improvement of results we have obtained for a F2F tracking operator by adding the appropriate knowledge. A true positive link is a link created by the tracking process combining two bounding boxes that both sufficiently cover a ground truth object at times t and t+1. All links made by the tracking process which are not true positives are classified as false positives. A false negative link is a link missed by the tracking process. For easy sequences, we measured 100% of true positives and 1% of false positives using the supervised evaluation. For the most challenging data previously mentioned, we report 88% of true positives, 12% of false negatives and 2% of false positives as opposed to 75%, 25% and 8% respectively before. Nevertheless, we have to point out that it can be difficult even for a video processing expert to find the correct piece of knowledge to add. Each improvement requires a deep understanding of the problem and sometimes several iterations on the problem are needed.

4. Scenario modelling knowledge representation

In the previous section, we have presented the operator formalism combined to a technical evaluation to improve results. At this point, the video understanding process is robust enough. Nevertheless, we must ensure that this solution is useful and that it addresses a real-life problem. This is the topic of this section.

4.1. Domain expert knowledge and end-user interaction

In order to clearly understand end-user needs and to acquire knowledge on a target application domain, we have conducted an interactive process for knowledge acquisition. It is an incremental process composed of three main steps:

- User need description: users explain their concrete need and they give us details on the target application domain: what to recognize and why it is important for them. We can distinguish end-users and experts. End-users are typically video surveillance human operators who bring us details about a scenario (e.g., what are important actions, how to detect it visually). Experts bring us other types of details and specify final goals (e.g., consequences or impact on customers, cost caused by robberies, typical robber profiles).

- Representation of these needs into scenario models: this step is of prime importance since it is a way of capitalizing the knowledge expressed by experts and to have a common representation which serves as a base for discussions. We try to determine visual invariants which will guide the recognition process and link video processing operators to end-user needs.

- Validation with end-users: verification that recognized scenarios correspond to what has been described in the description step. If it is not sufficient or adequate, the process is reiterated and a more precise description is given. The whole process is reiterated until the validation step gives good results.

For instance, we list hereunder knowledge samples given by bank experts. Today, classical bank agencies gradually evolve towards agencies with one or several counters without money, ATM (Automatic Teller Machine), safe room, and offices for commercial employees. The safe room is then the most significant zone since all the money is stored inside. As a consequence, all irregular behaviours or bank protocol infringement (involving either robbers or maintenance and cleaning employees) must be detected nearby the safe entrance. This protocol can be different for each bank. For instance, a rule specifies that only one person can enter the safe room at a time. We can distinguish different attack types:

- Counter attack: frequent, often stealthy, rapid and hardly observable even for human beings. The bank employee is threatened but it is generally difficult to see the difference with a classical customer request.

- Safe attack: not frequent. Bank employees and customers are threatened. People are traumatised and things can take a bad turn.

- Aggressive attack: bank employees and customers are threatened. The robber has lost his/her self control, money is not the main motivation and the robbery usually leads to a drama.

4.2. Scenario representation formalism

To represent this application domain knowledge, we have used the formalism of [12] which includes a language

to describe scenarios. A scenario can be of different types and composed of states and events. A state is a spatio-temporal property defined at one time instant or on a time interval. An event is one or several change(s) of states at two successive time instants or on a time interval. Scenarios can be either primitive (single state change) or composite (combination of states and events). They are described by three parts: *physical objects* (real world objects present in the scene), *components* (list of states and events involved in the scenario) and *constraints* (relations between physical objects and/or components). For instance, for a bank monitoring application, physical objects can be of two different types:

- Mobile objects as people or group of people (robber, customer, kid, director or cleaning employee) and portable objects (suitcase, stroller or gun).

- Contextual objects as predefined zones (entrance, back_counter, infront_counter, safe, safe_entrance) and equipment (counter, chair, desk, ATM, safe gate).

When constructing the scenario model library, we first select a set of primitive states and events. Composite events are then defined using this primitive set. An example of each one is given in Fig. 4. In a second time, we can build more complex scenarios, which are a combination of primitive and/or composite events. Currently, we have defined the following scenarios containing 1 to 3 persons (robber, bank employee, customer). These persons move relativley to five zone types and interact with the safe gate equipment:

- 1 person: the bank employee is behind or in front of the counter and goes to the safe. Then, the safe gate is opened.

- 2 persons: the employee is behind the counter. The robber enters the bank, goes to the counter and stays in front of it. Both people go to the safe.

- 3 persons: the bank employee is behind the counter. A customer enters the bank agency, goes to the counter and stays in front of it. After, a robber joins the customer. The employee and the robber go to the safe and the safe gate is opened. The customer stays behind the counter or leaves the agency.

4.3. End-user evaluation

There are two end-user evaluation types: based on recorded video sequences (several hours annotated by end-users) and on live video streams (two live evaluations performed inside a bank agency during one hour with end-users). The first evaluation produced average results and some scenarios were not recognized. This was due to unclear specifications and to the fact that end-users have refined their objectives when viewing a demonstration of the system. Using the description language, we thus refined scenario models together with experts and we identified two parts in a bank scenario: the attack precursor (i.e., the robber approach) and the attack. For bank experts, the attack (number of people inside the safe) must be recognized with a very high confidence. The attack precursor is facultative for bank attack detection but important in order to anticipate potential actions and prevent any drama. The second evaluation produced good results. A true positive is an alert raise when a real attack happens (simulated by actors), a false negative is the miss of an alert and a false positive is an alert raise for no real attack. The bank_attack scenario with 3 persons was played 16 times. We obtained 93.75% of true positives, 6.25% of false negatives and 0% of false positives when 2 people enter the safe room. The scenario with 2 persons was played more than 10 times and we obtained 100% of true positives. Such results has been reached thanks to the underlying methodology of interaction with domain experts. In addition, it is important for application purpose to reach such a false alarm rate. Most end-users would actually switch off a system which raises too many false alarms and would not trust it anymore.

5. Conclusion and future work

In this paper, we have presented a methodology to conceive efficient real-world semantic video understanding systems. This methodology combines knowledge representation formalisms with an evaluation framework to acquire and make the best use of all the available knowledge. We have insisted on the importance of having three properties for knowledge representation: isolated, modular and upgradable. For video processing programs, the operator formalism and an automatic technical evaluation enable a video processing expert to add the missing knowledge. Although this methodology requires a complete understanding of the processing, we report interesting improvement of results. A second formalism for scenario modelling combined with an end-user evaluation has shown to be an adequate way to understand end-user needs and address real-world problems. The scenario formalism serves as a shared language when discussing with experts and as a repository of the application domain knowledge. An evaluation of a bank monitoring application was carried out in live conditions with the presence of experts. First, end-users were able to play all scenarios in live and check directly the recognition results. Second, experts were able to easily refine scenario models and better express their expectations thanks to the scenario formalism. The evaluation has thus formally validated the global approach especially through the back and

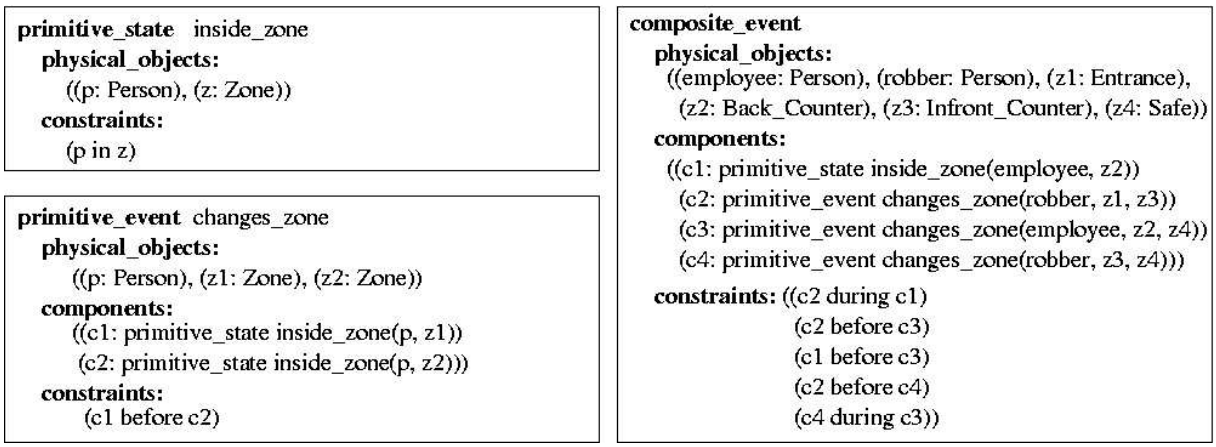

Figure 4. Upper left: primitive state model, A person p is inside zone z. Bottom left: primitive event model, A person p goes from zone z1 to zone z2. Right: composite event with 2 persons using primitives on the left. It corresponds to the scenario with 2 persons.

forth interaction with experts. Future work will investigate how we can further automate improvements especially by the use of learning techniques.

References

[1] J. Bins, T. List, R. Fisher, and D. Tweed. An intelligent and task-independent controller for video sequence analysis. In *Proceedings of the IEEE International Workshop on Computer Architecture for Machine Perception (CAMP05)*, pages 172–177, Palermo, Italy, July 2005.

[2] F. Brmond and M. Thonnat. Issues of representing context illustrated by video-surveillance applications. *International Journal of Human-Computer Studies Special Issue on Context*, 48(3):375–391, 1998.

[3] X. Desurmont, C. Chaudy, A. Bastide, J.-F. Delaigle, and B. Macq. A seamless modular image analysis architecture for surveillance systems. In *Proceedings of the 2th Workshop on Intelligent Distributed Surveillance Systems (IDSS'04)*, pages 66–70, London, United Kingdom, February 2004. IEE London.

[4] T. Ellis. Performance metrics and methods for tracking in surveillance. In J. Ferryman, editor, *Proceedings of the 3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS'02)*, pages 26–31, Copenhagen, Denmark, June 2002.

[5] J. Ferryman, editor. *IEEE International Series of Workshops on Performance Evaluation of Tracking and Surveillance (PETS)*. IEEE Computer Society, 2002. http://visualsurveillance.org.

[6] I. Haritaoglu, D. Harwood, and L. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[7] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, 1996.

[8] C. LeGal, J. Martin, A. Lux, and J. Crowley. Smartoffice: Design of an intelligent environment. *IEEE Intelligent Systems*, 16(4):60–66, 2001.

[9] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modelling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[10] T. Schoepflin, C. Lau, R. Garg, D. Kim, and Y. Kim. A research environment for developing and testing object tracking algorithms. In *Proceedings of the SPIE, Electronic Imaging 2001*, pages 667–675, 2001.

[11] M. Thonnat. Knowledge-based techniques for image processing and for image understanding. *J. Phys. IV France EDP Science, Les Ulis*, 12(1):189–236, 2002.

[12] T. Vu, F. Brmond, and M. Thonnat. Automatic video interpretation: a recognition algorithm for temporal scenarios based on pre-compiled scenario models. In J. Crowley, J. Piater, M. Vincze, and L. Paletta, editors, *Proceedings of the 3rd International Conference on Computer Vision Systems (ICVS'03)*, Lecture Notes in Computer Science, pages 523–533, Graz, Austria, April 2003. Springer-Verlag.