Chapter 12

Activity Recognition

12.1. Introduction

Activity recognition is a key stage in automatic analysis of video sequences. The main challenge is to make the link between low-level input and a semantic description of the activities. The problem is complex because of the noise-afflicted nature of the videos (lighting changes, segmentation problems, occlusions, etc.), which provide incomplete, or even erroneous, low-level information. The interest of researchers in activity recognition is also due to the great many possible applications for smart systems: smart video surveillance, video indexing, home care for the elderly, etc. The approaches used for activity recognition from videos are usually classified into two broad categories: probabilistic approaches and constraint-based approaches. However, we can see that this classification is unable to account for all the diversities of the different recognition techniques. In this chapter, we focus, in particular, on the representation of knowledge, modeling of scenarios by the users and automatic recognition of these scenarios. We illustrate our proposal by presenting an activity recognition system: scenario recognition based on knowledge (ScReK). We show how to introduce probabilistic activity recognition using the proposed method of analysis of the spatiotemporal constraints. Finally, this method is illustrated on two applications: analysis of the behavior of elderly people and monitoring of activities around an airplane on the tarmac at an airport.

Chapter written by Bernard BOULAY and François BREMOND.

12.2. State of the art

Activity recognition from video sequences is a difficult problem because of the noisy nature of the videos and the uncertainty of the results of algorithmic treatments (e.g. detection and tracking of objects of interest). Typically, activity recognition takes a video sequence as input and extracts from it any objects of interest for a given application: this stage is called abstraction. Then, these objects of interest are used in the process of activity recognition. It is important, in this review of the state of the art, to discuss the type of data input for activity recognition and the methods for modeling and recognition of the activities themselves. Thus, we first present the different levels of abstraction possible for input to the recognition algorithms, before detailing on the various techniques for modeling and recognition of activities.

12.2.1. *Levels of abstraction*

Abstraction represents images by a set of representative objects of interest. This stage is essential because it defines the activity modeling and recognition techniques which will be applicable.

– The first possible abstraction is extracted from the characteristics of a pixel or a set of pixels such as the color, texture and the gradients. An example that is frequently used for activity recognition is the motion history image [BOB 01], which shows the history of the moving pixels in relation to a reference image and calculated over a time window.

– Another possibility is object-based abstraction. Indeed, representing a sequence by the objects that make it up is a good alternative in order to recognize activities. The objects may, for example, be people or vehicles, with different associated properties such as the velocity or trajectory. In the literature in this field, many approaches use this level of abstraction [OLI 00, VU 03] because it is easy to model the activities naturally.

Comment [A1]: AQ: Please check the usage of [naturally] in the text and provide a suitable alternative if possible.

– A final possible level of abstraction uses logic: the representative objects extracted from the low-level data are associated with a semantic concept. These notions can then be used by way of logical rules to model the activities.

There are times when a certain approach does not appear to obviously fit into a certain level of abstraction, but we believe that this classification provides an overall view of the techniques currently available. For instance, the approach described in [ROM 10] uses people (object-based abstraction) detected by segmentation and tracking as input for activity recognition. Yet the activities are modeled according to different levels. The first level, called the primitive event, corresponds to logic-based abstraction.

12.2.2. *Modeling and recognition of activities*

Activity modeling and recognition techniques are usually classified into two categories: probabilistic approaches and deterministic approaches. Lavee *et al.* [LAV 09] propose another classification method that better accounts for the diversity of existing techniques. The approaches are classified into three categories:

Comment [A2]: AQ : Does [the approaches] refer to the two approaches said previously? If so, it should be [each of the approaches is classified….]. Please suggest.

– methods using a pattern recognition model;

– state-based models;

– semantic models.

Both the latter two models use semantic information, but it is helpful to differentiate state-based approaches (classically hidden Markov models – HMMs) from models that explicitly express the temporal relations between subactivities more naturally.

12.2.2.1. *Methods using a pattern recognition model*

Methods that use a pattern recognition model are not necessarily approaches developed specifically to recognize events but are conventional recognition techniques using a classifier. The main advantage is that these techniques are clearly defined and have been used for a long time. The drawback is that the semantics of the domain is used only at a high level, directly at the level of the classifier. More often than not, the addition of new types of activity requires that these classifiers be computed anew. Examples of this include nearest neighbor methods, boosting techniques, support vector machine (SVM) and neural networks [BAR 03, CHE 06].

12.2.2.2. *State-based models*

State-based models enable us to formalize the activities in terms of space and time using semantic knowledge. Each state represents a significant part for one activity. These techniques are well formulated mathematically. Learning the different parameters is often a difficult and painstaking process, but once the model (the semantics) is specified, it becomes possible for the system to learn from a data set. State-based models include:

– finite state machines (FSMs);

– Bayesian networks (BNs);

– hidden Markov models (HMMs);

– dynamic Bayesian networks (DBNs);

– conditional random fields (CRFs).

FSMs model the activities sequentially. The models thus defined are simple in terms of relations between the states. FSMs are clearly formulated and therefore enable the recognition algorithms to be optimized. They have been used, for instance, to recognize interactions between several people [HON 01]. A number of attempts have been made to introduce a concept of uncertainty into these models, but they are too specific to a particular application to be generalized. Furthermore, there are models that are better adapted, such as HMMs.

BNs offer a solution to take account of the uncertainties that exist for activities recognized in video sequences. Usually, BNs model the activities as a binary variable (either the activity has occurred or it has not) and the observations as the known variables. The main advantage of BNs [OLI 05] is that it is possible to model the uncertainty of recognition using probabilities based on Bayes' theorem. For instance, Chomat and Crowley [CHO 99] address the issue of a probabilistic recognition of activities (such as "a person walking") and hand gestures using the local spatiotemporal appearances and Bayes' principles. The major disadvantage of BNs is that it is impossible to model the temporal compositions (while, during, etc.). In addition, the *a priori* probabilities have to be learned, and this stage is often difficult. It requires a data set representative of the domain under examination.

HMMs [HOE 07, NEV 04] combine the advantage of FSMs – modeling of the evolution over time – with that of BNs – probabilistic modeling. However, due to the intrinsic nature of HMMs, complex temporal relations (e.g. "while") are not easy to model. In addition, the modeling of activities involving multiple objects is limited. Indeed, the probability of an object being in a certain state must be combined with the probability of all other objects being in a different state. This combination leads to an exponential increase in the computation time for the recognition process. Thus, many versions of the HMMs have been put forward in order to be able to model and understand the complex scenes involving multiple objects. Oliver *et al.* [OLI 00] exploit coupled hidden Markov models (CHMM) to model basic interactions between people, such as "one person following another person" and "changing direction to meet another person". Xiang and Gong [XIA 06] presented a dynamically multi-linked hidden Markov model (DML-HMM) to model outdoor activity. The DML-HMM topology was constructed due to the discovery of representative dynamic interlinks from complex activities. Duong *et al.* [DUO 05] suggest using a two-layer switching hidden semi-Markov model (S-HSMM) to recognize a series of activities. The lower layer represents basic activities and the upper layer represents a series of complex activities made up of combinations of basic activities. The work of Liu and Chua [LIU 06] proposes observation decomposed hidden Markov models (ODHMM) to model multi-agent activities. Yet the disadvantage of all these extensions of the HMMs is that because the model

becomes more complex, the conventional recognition algorithms no longer apply and leave room for approximations.

Dynamic BNs (DBNs) generalize BNs, adding a function to handle temporal relations. They have been successfully tested, recognizing short temporal actions. However, the process of recognition depends on the duration of the activity: when the frame rate of the video or the duration of the actions changes, the DBN needs retraining.

The main drawback of these different models is the need to know *a priori* information (such as the probabilities). Such information is rarely known and is, therefore, estimated with certain conditions and simplifications. The most widely used hypothesis is that of independent states, which is rarely the case for activities. Therefore, CRFs [LAF 01] generalize HMMs to avoid making this kind of simplification.

12.2.2.3. *Semantic models*

Semantic models define the relations, which may be spatial or temporal, between subactivities to form a more complex activity. Because of the nature of these models, the activities must be defined by an expert in the field of application. It is indeed difficult to learn these models. Furthermore, these models are often deterministic, and probabilistic recognition is difficult or even impossible. A number of techniques have been studied:

– grammar-based models;

– Petri nets (PNs);

– constraint-based models and logical approaches.

Description of the activities in a video in human language has led to the use of stochastic grammar to analyze simple actions. Typically, a grammatical model comprises a set of terminations (the abstract features), a set of non-terminations (the subactivities) and a set of rules (the structure of the activity). In [KIT 07], the authors propose an approach to automatically learn a grammatical model to represent the activities in a video.

PNs enable us to model temporal relations (sequential and non-sequential) constituting the activities. The nodes of the net may either be the objects of interest in a video (we then speak of an object-oriented PN) or subactivities (we then speak of a flat PN). The transition nodes are either a change in an object or a solution of a constraint. The main advantage of PNs is the possibility of recognizing incomplete events. The main drawback of these nets is their deterministic nature.

Constraint-based approaches have also been widely used for activity recognition. Their main characteristic is that they define a symbolic net wherein each node or predicate corresponds to the binary recognition of a simple activity. Constraint satisfaction problems (CSPs) have been applied to model the activities as a network of constraints [RED 09, VU 06, ZOU 09]. For instance, Ghallab [GHA 96] represents an activity as a set of temporal constraints on activities that contain dates (time stamps). The recognition algorithm propagates the time stamps using the Rete algorithm. The approach proposed by Vu *et al.* [VU 03] also uses a declarative representation of the activities, defining a set of spatiotemporal and logical constraints. The major problem with these approaches is the difficulty in modeling uncertainty in activity recognition.

Note also that programming languages such as *Logic* and *Prolog* have been used for recognition of activities defined as predicates [SHE 05]. These approaches are useful when the number of predicates is low, which corresponds to a small number of objects in the video.

12.2.3. *Overview of the state of the art*

As we have just seen, there are a multitude of techniques to represent and recognize activities in video sequences. In addition, there are many possible abstractions that can be used to forge the link between the video world and the real world in which the activity takes place. We believe that knowledge of the domain is important in order to be able to correctly interpret a scene. Furthermore, because of the heavily noised nature of videos, purely deterministic recognition has clear limitations. We will now discuss ontology and representation of knowledge. Then, we will present the ScReK system [ZAI 11], which offers various tools to aid in the construction of an ontology and which enables systems to recognize probabilistic activities by solving spatiotemporal constraints.

12.3. Ontology

Representation of the knowledge in the domain of application is crucial for activity recognition. Indeed, knowledge of the domain is important in order to be able to lend a semantic dimension to algorithmic detections. This knowledge (concepts and properties) is often represented using Web Ontology Language (OWL) or Semantic Web Rule Language (SWRL). Inference engines enable us to use the SWRL: Bossam, Pellet, KAON2 [KOL 06]. The advantage of these languages is that they also have algorithms to verify the coherence of the models defined. However, from our point of view, the major disadvantage is that this

formalism is not easy to use, and although there are graphic tools such as Protégé[1], a consummate expert in a particular field would have difficulty in defining the ontology of his/her field without an in-depth knowledge of the mechanisms of the language. However, we believe it is important that a user be able to model the knowledge in his/her domain him/herself, using a tool and/or a simple language. For this reason, in ScReK, we offer a language that can be used to describe the different parts of an ontology needed for activity recognition based on video sequences.

An ontology requires the description of three different kinds of knowledge:

– the objects of interest;

– the scenario models;

– the operators.

12.3.1. *Objects of interest*

Objects of interest may be the objects detected in the video-by-video analytic algorithms. These are dynamic objects, which vary over the course of time. New ones may appear; they may disappear and their attributes evolve frequently over time. However, they may also be pieces of *a priori* knowledge of the scene: how is the furniture laid out? What are the zones of interest? These are static objects, which are constant over time. They define the context of the scene. These objects will differ from one domain of application to another. However, they can be reused, either in their entirety or in part. For instance, if in an application for posture recognition, the object of interest is a person, that object can be reused in the domain of surveillance of a parking lot. This is one of the advantages of using an ontology. It can be reused for another domain. An object is made up of a name and a set of attributes. These attributes have a name, which can be used in the scenario models, and a type, which is defined in a set of basic concepts. These basic concepts may be classic in nature (integers, strings, etc.) or more complex (a 3D point, a time interval, etc.). ScReK offers 10 basic concepts (time stamp, interval, Boolean, etc.). The set of concepts is open-ended and new concepts may be added to cater for the users' needs. The idea is that the more the system is used, the more complete the basic concepts will be. An example of the modeling of an object of interest is shown in Figure 12.1. The description language is very simple, and users can model their own objects of interest in a matter of minutes. The language also enables hierarchical modeling: an object inherits attributes from its parent.

1 Protégé, an open-source OWL ontology editor, http://protege.stanford.edu/

```
class Mobile {
const false;
CS2DDPoint Position2D;
CS3DDPoint Position3D;
CSDouble Width;
CSDouble Height;
CSDouble Depth;}

class Person:Mobile{
const false;
CSInt Posture;}
```

```
class ContextualObject {
const true;
CSString Name;
CS3DDPoint Position;}


Class Zone:ContextualObject {
const true;
CS3DDPoints Vertices; }
```

Figure 12.1. *Example of the definition of some objects of interest. The mobile object is a dynamic object, which can vary over time. It comprises a 2D position, a 3D position, a width, height and depth. The object Person inherits the properties of Mobile (Person objects are Mobile objects) and also includes a posture. The object ContextualObject is defined as constant because it does not vary over time*

12.3.2. *Scenario models*

The scenario models that define the activities to be recognized are the second piece of knowledge that must be defined. These models use the objects defined previously. It is important to correctly model the layer that forms the link between the video world and the world of the activities. To this end, the activities are classified into four layers, from lowest (nearest to the vision algorithms) to highest (the activities):

– Primitive state is a spatiotemporal property, valid at a given time or stable over a period of time, which is directly computed based on an attribute of an object of interest. This layer forms the link between the world of the detections (abstraction layer) and that of the activities.

– Composite state is a combination of states.

– Primitive event is a change in state.

– Composite event is a combination of states and events. Usually, the composite event relates directly to the domain in question. It generally represents the activities that are easy to recognize.

Comment [A3]: AQ: The sentence [It generally represents….] OK as edited?

A scenario is made up of five parts:

– Physical objects, which are the objects of interest involved in the scenario.

– The components are the subactivities that define the scenario.

– The constraints describe the relations between the components (temporal constraints), between the objects (spatial constraints) or on the objects' attributes (logical constraints). The constraints are written using an operator or a combination of operators. A description of the operators is given in the next section.

– The alarm describes the importance of the scenario for the desired application. This part is not involved in recognition. It is used, e.g., to control what is displayed to the user.

– The action gives a function to perform if the scenario is detected. This function may be purely algorithmic, or might give a command to the hardware (zoom and movement of a camera, for example).

Examples of different scenarios can be seen in Figure 12.2.

```
PrimitiveState(Person_inside_Zone,
PhysicalObjects( (p : Person), (z : Zone))
Constraints((p in z))
Alarm((Level : NOTURGENT)))

CompositeEvent(AFT_CN_LoadingUnloading_Ends,
PhysicalObjects( (v1 : Vehicle), (v2 : Vehicle), (z1 : Zone), (z2 : Zone) )
Components(     (c1 : AFT_CN_LoadingUnloading_Operation_Ends(v1,v2, z1,z2))
                (c2 : AFT_LD_Removing(v1, z1)) )
Constraints( (c1 before c2) )
Alarm ((Level : URGENT)))
```

Figure 12.2. *Examples of scenario models*

In these examples:

– The primitive state Person_inside_Zone takes two objects of interest as input, a person and a zone, and verifies that the position of the person p is indeed inside the zone z.

– The composite event AFT_CN_LoadingUnloading_Ends verifies the temporal constraint "before" between the two subactivities it contains.

12.3.3. *Operators*

The operators are operations used in the constraints of the scenario models. We believe it important that they form part of the ontology. Indeed, the definition of an operator may vary from one application to another. For instance, if we consider the operator "in", which verifies whether an object is in a zone, a number of definitions

are possible. Either the position of the object is denoted in the form of a point or the position of the object is defined by the surface that it occupies on the ground. In these two cases, the calculation of "in" will be completely different. We define three types of operators:

– Basic operators use one or more attributes and return a value.

– Historical operators use one or more attributes and their historical values over time and return a value.

– Probabilistic operators use one or more attributes and return a value and its associated uncertainty. Probabilistic operators enable non-deterministic activity recognition.

The ScReK system offers 17 operators, of temporal, spatial and logical types (see Table 12.1). The operators are generic: for instance, equality (==) can be used with all the basic concepts.

Type of operators	
Temporal	Before, And, Meet, Overlap
Spatial	Distance, In, Direction, Speed
Logical	!=, ==, <, <=, >, >=, !(not), Duration, DurationBetween

Table 12.1. *List of the different operators available in ScReK*

12.3.4. *Summary*

In this part, we have presented the different pieces making up the ontology necessary for activity recognition in video sequences. The next section describes how this knowledge is used in recognition.

12.4. Suggested approach: the ScReK system

The scenarios are modeled using spatiotemporal constraints. The ScReK system offers a generic constraint verifier (see Figure 12.3) that facilitates activity recognition. The verifier is generic because the operators are defined based on the attributes of the objects (the basic concepts) rather than directly on the objects themselves. The same recognition engine can therefore be applied for different domains of application. In addition, the verifier is able to handle different types of abstraction in the video sequence. In what follows, we will use object-based abstraction to illustrate our points.

The input to the verifier consists of the objects detected by a video processing platform, which instantiate the scenario models. The constraints of these instances are then verified to see whether or not the recognition takes place. Finally, the recognized instance is compared with the previously recognized instances to discover whether the activity has already been recognized previously (merging) or whether it is new (creation).

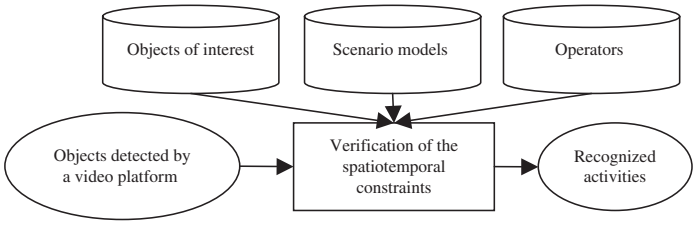

Figure 12.3. *Overall view of the constraint verifier in ScReK. The cylinders correspond to the modeled knowledge. The verifier inputs the objects detected by video processing algorithms and outputs the recognized activities*

If all the models are instantiated at all times, the processing time will explode with an increasing number of scenarios and of detected objects. To limit the computation time, recognition is only performed for plausible scenario models. ScReK takes as input optimal scenario models. Such scenarios are composed of at most two subscenarios (at most two components) and at most one temporal constraint. There may be as many spatial and logical constraints as required. This property is not restrictive because any scenario can be modeled in this form. Based on these optimal models, the scenario model tree is created. The tree defines which subscenario can trigger the recognition of the overall scenario: the subscenario in question is that which happens last. For instance, scenario A has two components: B and C. The temporal constraint is "B before C". In this case, recognition of C would trigger the recognition of A because C occurs after B. Thus, using a scenario tree noticeably limits the computation time by only activating the recognition of plausible scenarios. The recognition algorithm comprises three stages:

– attempting to recognize all possible simple scenario models (primitive states), by instantiation of all the models with the objects of interest available at time *t*;

– attempting to recognize complex scenarios based on the scenario tree and the simple scenarios recognized previously;

– verifying whether the scenario recognized at time *t* has already been recognized previously in order to update it (end of the event) or to create a new one.

As discussed above, the objects detected are noised because they come from the treatment of video sequences. To take account of these errors, probabilistic operators can be used instead of the deterministic operators. These operators are particularly important at the level of the primitive states, which are the link between the video and the real-world events. They enable us to recognize simple events that would not otherwise have been recognized and therefore would not have triggered the recognition of more complex events. Here, we use a method proposed by Romdhane *et al.* [ROM 10], who use a Bayesian method to calculate and propagate the uncertainties. The probability of a complex event ω at time t depends on the probability of its subevents SE_i^0 and the probability associated with its constraints $\zeta_{e_c}^O$:

$$P\left(\omega = e_c \mid SE_i^O\left(e_c\right) = se_i, \zeta_{e_c}^O = \text{true}\right) = P\left(SE_i^O\left(e_c\right) = se_i \mid \omega = e_c\right) * \qquad [12.1]$$

$$P\left(\zeta_{e_c}^O = \text{true} \mid \omega = e_c\right) * \frac{P\left(\omega = e_c\right)}{P\left(SE_i^O\left(e_c\right) = se_i\right).P\left(\zeta_{e_c}^O = \text{true}\right)}$$

The last term in the equation represents the marginal probabilities of SE_i^0, $\zeta_{e_c}^O$ and ω. It plays the role of a normalizing constant.

The default operator *in* offered in ScReK verifies whether a point P is inside a set of points. The probabilized operator *in* looks at the distance from point P to the boundaries of the set of points. The closer the point is, the greater the associated probability will be. This calculation enables us to recognize new primitive states compared to the deterministic *in* and thus to recognize new activities.

Comment [A4]: AQ: Please rephrase the sentence [The probabilized operator….] for clarity.

In the next section, we illustrate the use of ScReK for two applications in the domain of an airport and a hospital.

12.5. Illustrations

In this part, the abstraction layer used is object-based. A video processing platform feeds into ScReK a set of objects detected, classified and tracked.

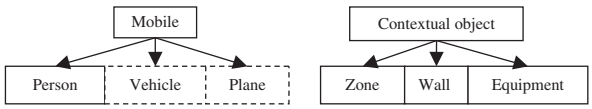

Figure 12.4. *Modeling of objects of interest. In solid boxes, the objects are common to the hospital and the airport. In dotted boxes, the objects are specific to the airport*

Figure 12.5. *Example of the activity of baggage loading being performed near an airplane in the context of the Co-Friend project (left). Example of a test being performed at the CHU (University Hospital) in Nice (right). A patient has to carry out a series of physical exercises, under the guidance of the medical staff*

12.5.1. *Application at an airport*

The European project Co-Friend[2] aims to model the activities occurring in the vicinity of an airplane at rest on the airport apron (see Figure 12.5). The activities in the airport are described very precisely because they follow a predefined protocol (for instance, the ground power unit (GPU) vehicle must be in position before the aircraft arrives).

Comment [A5]: AQ : Probably would be better to say [airport ramp]?

The objects of interest for this application are visible in Figure 12.4. A little more than 80 scenario models have been defined to represent the different activities around the plane. Over half of these models are independent of the domain of application (e.g. Person_inside_zone, Vehicle_stopped). The operators used in the modeling of the constraints are the same operators provided by ScReK. The results of recognition of interesting activities (i.e. which are helpful for the user – indeed, activities such as Vehicle_Inside_Zone are of no consequence) are shown in Table 12.2. A complete turnover corresponds to the period from arrival of the aircraft to its departure, which is, on average, a little more than an hour.

12.5.2. *Modeling the behavior of elderly people*

As part of the French ANR project Sweet-home[3], in cooperation with the CHU in Nice, the *Stars* team at INRIA (French National Institute for Research in Computer Science and Control) has devised a non-invasive system using video cameras to help form a model of the behavior of elderly people. The people are

2 CoFriend, European project, available at www.co-friend.net/
3 Sweet-home, ANR project, available at http://cmrr-nice.fr/sweethome/

asked to perform various physical exercises in accordance with a protocol proposed by the doctors (see Figure 12.5). The ScReK system is used to recognize activities such as getting up, sitting down rapidly, walking in a specific area, and so on.

	True positive	False positive	False negative
Total	18	29	9
GPU_Positioning (1)	3	5	2
PBB_Positioning (2)	4	10	1
Fwd_Loading/Unloading (3)	4	7	0
Aft_Loading/Unloading (4)	5	3	4
PB_Positioning (5)	2	4	2

Table 12.2. *Results of activity recognition for five complete turnovers in the airport application: (1) positioning of the GPU vehicle GPU; (2) positioning of the passenger boarding bridge; (3) loading and unloading of baggage at the front; (4) loading and unloading of baggage at the back; and (5) positioning of the Push-Back vehicle (the vehicle which pushes the airplane onto the runway)*

The ultimate goal is to be able to model profiles of healthy patients and patients suffering from Alzheimer's disease. The objects of interest for this application can be seen in Figure 12.5. The ontology used in the domain of the airport has been reused. In addition, the models of the primitive scenarios are the same as for the airport (Person_inside_zone, Person_stopped, etc.). These models have been supplemented with scenarios regarding the people's posture, and scenarios specific to the hospital (e.g. Up-Go – the patient gets *up* from his/her chair and *go*es away). In addition, probabilistic operators have been added (*in*, etc.) An example of recognition with and without the use of probabilistic operators is given in Table 12.3. The use of uncertainties enables us to better recognize the sought activities (for instance the activity *up-go* is recognized on 59.2% of occasions in the deterministic case, and 92.6% in the probabilistic case).

In general, the effort required to switch from one domain of application to another is minimal, so long as the abstraction layer is the same in both cases.

Events	No. of videos	No. of actors	TP	FP	FN
Walking exercise	27	1	16/**25**	3/**5**	11/**2**
Start of the test	9	2	8/**9**	1/**1**	1/**0**
Interaction with a chair	10	1	10/**10**	0/**0**	1/**0**

Table 12.3. *Comparison of the rate of true positives (TP), false positives (FP) and false negatives (FN) in the deterministic and probabilistic (shown in bold) cases*

12.6. Conclusion

In this chapter, we have given a topical outline of the different techniques for activity recognition from video sequences.

We have seen that it is important to describe the abstraction layer because it guides the choice of an activity modeling and recognition technique. If the user's need is to be able to automatically detect activities in a video stream, the ideal level of abstraction must, in our view, be pixel based [PUS 11]. We believe that if the goal is to create a generic recognition system, the abstraction must be object based because the objects can be directly comprehended by an expert in the domain, and the activities will be modeled naturally.

Furthermore, we have seen that it is necessary to properly separate the knowledge from the recognition itself. An ontology can be reused for different domains of application. It could also be used with other recognition systems. The ScReK system presented herein satisfies these criteria and enables users to easily model the ontology of an application. Furthermore, the recognition engine (the constraint verifier) is generic and could be used for other types of abstraction.

We saw the need to model the context of the scene and, in particular, the important zones. It is interesting to be able to model these zones automatically. The project described in [PAT 10] works toward this direction and enables us to automatically learn the zones of interest by analyzing the trajectories of the objects detected. The next step is to automatically learn the primitive states based on these same data.

12.7. Bibliography

[BAR 03] BARNARD M., ODOBEZ J., BENGIO M., "Multi modal audio-visual event recognition for football analysis", *IEEE Workshop on Neural Networks for Signal Processing*, Toulouse, France, 2003.

[BOB 01] BOBICK A., DAVIS J., "The recognition of human movement using temporal templates", *IEEE Transactions on PAMI*, vol. 23, no. 3, pp. 257–267, 2001.

[CHE 06] CHEN X., ZHANG C., "An interactive semantic video mining and retrieval platform-application in transportation surveillance video for incident detection", *6th IEEE International Conference on Data Mining (ICDM '06)*, Hong Kong, China, 2006.

[CHO 99] CHOMAT O., CROWLEY J., "Probabilistic recognition of activity using local appearance", *CVPR*, Fort Collins, United States, 1999.

Comment [A6]: AQ : Please provide the expansion of the abbreviated conference title in [CHO 99], [DUO 05], [HON 01], [KIT 07], [NEV 04], [PAT 10], [PUS 11], [ROM 10], [RED 09], [SHE 05], [VU 03] and [VU 06].

[DUO 05] DUONG T.V., BUI D.Q., PHUNG D.Q., VENKATESH S., "Activity recognition and abnomality detection with the switching Hidden Semi-Markov Models", *CVPR*, San Diego, CA, 2005.

[GHA 96] GHALLAB M., "On chronicles: Representation, online recognition and learning", *5th International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, vol. 5, no. 8, pp. 597–606, 1996.

[HON 01] HONGENG S., NEVATIA R., "Multi-agent event recognition", *ICVS*, Vancouver, Canada, 2001.

[HOE 07] HOEY J., VON BERTOLDI A., POUPART P., MIHAILIDIS A., "Assisting persons with dementia during handwashing using a partially observable Markov decision process", *ICVS*, Bielefeld, Germany, 2007.

[KIT 07] KITANI K., SATO Y., SUGIMOTO A., "Recovering the basic structure of human activities from a video-based symbol string", *IEEE WMVC*, Austin, TX, 2007.

[KOL 06] KOLOVSKI V., PARSIA B., SIRIN E., "Extending SHOIQ(D) with DL-safe rules: first results", *International Workshop on Description Logics (DL '06)*, Windermere, Lake District, UK, 2006.

[LAF 01] LAFFERTY J., MCCALLUM A., PEREIRA F., "Conditional random fields: probabilistic models for segmenting and labelling sequence data", *ICML*, Williamstown, MA, 2001.

[LAV 09] LAVEE G., RIVLIN E., RUDZSKY M., "Understanding video events: a survey of methods for automatic interpretation of semantic occurences in video", *IEEE Transactions on SMC – Part C: Applications and Reviews*, vol. 39, no. 5, 2009.

[LIU 06] LIU X., CHUA C.S., "Multi-agent activity recognition using observation decomposed hidden Markov models", *Image and Vision Computing*, vol. 24, 2006.

[NEV 04] NEVATIA R., HONGENG S., BREMONF F., "Video-based event recognition: activity representation and probabilistic recognition methods", *CVIU*, vol. 96, no. 2, pp. 129–162, 2004.

[OLI 00] OLIVER N.M., ROSARIO B., PENTLAND A.P., "A Bayesian computer vision system for modeling human interactions", *IEEE Transactions on PAMI*, vol. 22, no. 8, 2000.

[OLI 05] OLIVER N., HORVITZ E., "A comparison of HMMs and dynamic Bayesian networks for recognizing office activities", *International Conference on User Modeling*, Edinburgh, UK, 2005.

[PAT 10] PATINO VILCHIS J.L., BREMOND F., EVANS M., SHAHROKNI A., FERRYMAN J., "Video activity extraction and reporting with incremental unsupervised learning", *AVSS*, Boston, MA, 2010.

[PUS 11] PUSIOL G., BREMOND F., THONNAT M., "Unsupervised discovery and recognition of long term activities", *ICVS*, Nice-Sophia Antipolis, France, 2011.

[ROM 10] ROMDHANE R., BREMOND F., THONNAT M., "A framework dealing with uncertainty for complex event recognition", *AVSS*, Boston, MA, 2010.

Comment [A7]: AQ : Please provide the page range for reference [LAV 09], [LIU 06], [OLI 00], and [ZOU 09].

[RED 09] REDDY S., GAL Y., SHIEBER S., "Recognition of users activities using constraint satifaction", *UMAP*, Trento, Italy, 2009.

[SHE 05] SHET V., HARWOOD D., DAVIS L., "Video monitoring of activity with prolog", *AVSS*, Como, Italy, 2005.

[VU 03] VU V.T., BREMOND F., THONNAT M., "Automatic video interpretation: a novel algorithm for temporal scenario recognition", *IJCAI*, Acapulco, Mexico, 2003.

[VU 06] VU V.T., BREMOND F., DAVINI G., THONNAT M., PHAM Q., ALLEZARD N., SAYD P., ROUAS J., AMBELLOUIS S., "Audio video event recognition system for public transport security", *IET ICDP*, London, UK, 2006.

[XIA 06] XIANG T., GONG S., "Beyond tracking: modeling activity and understanding behaviour", *International Journal of Computer Vision*, vol. 67, no. 1, pp. 21–51, 2006.

[ZAI 11] ZAIDENBERG S., BOULAY B., GARATE C., CHAU D.P., CORVEE E., BREMOND F., "Group interaction and group tracking for video-surveillance in underground railway stations", *ICVS- Workshop on Behaviour Analysis and Video Understanding*, Nice-Sophia Antipolis, France, 2011.

[ZOU 09] ZOUBA N., BREMOND F., THONNAT M., ANFONSO A., PASCUAL E., MALLEA P., MAILLAND V., GUERIN O., "A computer system to monitor older adults at home: preliminary results", *International Journal Gerontechnology*, SF-TAG: Gerontechnology- French Issue, vol. 8, no. 3, 2009.

Comment [A8]: AQ : The journal title has been spelt out in [XIA 06]. Please confirm.