Behaviour Recognition using the Event Calculus

Alexander Artikis and George Paliouras

Abstract We present a system for recognising human behaviour given a symbolic representation of surveillance videos. The input of our system is a set of time-stamped short-term behaviours, that is, behaviours taking place in a short period of time — walking, running, standing still, etc — detected on video frames. The output of our system is a set of recognised long-term behaviours — fighting, meeting, leaving an object, collapsing, walking, etc — which are pre-defined temporal combinations of short-term behaviours. The definition of a long-term behaviour, including the temporal constraints on the short-term behaviours that, if satisfied, lead to the recognition of the long-term behaviour, is expressed in the Event Calculus. We present experimental results concerning videos with several humans and objects, temporally overlapping and repetitive behaviours.

1 Introduction

We address the problem of human behaviour recognition by separating low-level recognition, detecting activities that take place in a short period of time — 'short-term behaviours' — from high-level recognition, recognising 'long-term behaviours', that is, pre-defined temporal combinations of short-term behaviours. In this paper we present our work on high-level recognition. We evaluate our approach using an existing set of short-term behaviours detected on a series of surveillance videos.

To perform high-level recognition we define a set of long-term behaviours of interest — for example, 'leaving an object', 'fighting' and 'meeting' — as temporal combinations of short-term behaviours — for instance, 'walking', 'running', 'inactive' (standing still) and 'active' (body movement in the same position). We employ the Event Calculus (EC) [6], a declarative temporal reasoning formalism, in order

Institute of Informatics and Telecommunications, National Centre for Scientific Research "Demokritos", Athens 15310, Greece
e-mail: `{a.artikis,paliourg}@iit.demokritos.gr`

Table 1 Main Predicates of the Event Calculus.

Predicate	Textual Description
happens(*Act*, *T*)	Action *Act* occurs at time *T*
initially(*F* = *V*)	The value of fluent *F* is *V* at time 0
holdsAt(*F* = *V*, *T*)	The value of fluent *F* is *V* at time *T*
holdsFor(*F* = *V*, *Intervals*)	The value of fluent *F* is *V* during *Intervals*
initiates(*Act*, *F* = *V*, *T*)	The occurrence of action *Act* at time *T* initiates a period of time for which the value of fluent *F* is *V*
terminates(*Act*, *F* = *V*, *T*)	The occurrence of action *Act* at time *T* terminates a period of time for which the value of fluent *F* is *V*

to express the definition of a long-term behaviour. More precisely, we employ EC to express the temporal constraints on a set of short-term behaviours that, if satisfied, lead to the recognition of a long-term behaviour.

The remainder of the paper is organised as follows. First, we present the Event Calculus. Second, we describe the dataset of short-term behaviours on which we perform long-term behaviour recognition. Third, we present our knowledge base of long-term behaviour definitions. Fourth, we present our experimental results. Finally, we briefly discuss related work and outline directions for further research.

2 The Event Calculus

Our system for long-term behaviour recognition (LTBR) is a logic programming implementation of an Event Calculus formalisation expressing long-term behaviour definitions. The Event Calculus (EC), introduced by Kowalski and Sergot [6], is a formalism for representing and reasoning about actions or events and their effects. We present here the version of the EC that we employ (for more details see [2]).

EC is based on a many-sorted first-order predicate calculus. For the version used here, the underlying time model is linear and it may include real numbers or integers. Where *F* is a *fluent* — a property that is allowed to have different values at different points in time — the term $F = V$ denotes that fluent *F* has value *V*. Boolean fluents are a special case in which the possible values are true and false. Informally, $F = V$ holds at a particular time-point if $F = V$ has been *initiated* by an action at some earlier time-point, and not *terminated* by another action in the meantime.

An *action description* in EC includes axioms that define, among other things, the action occurrences (with the use of the happens predicate), the effects of actions (with the use of the initiates and terminates predicates), and the values of the fluents (with the use of the initially, holdsAt and holdsFor predicates). Table 1 summarises the main EC predicates. Variables (starting with an upper-case letter) are assumed to be

universally quantified unless otherwise indicated. Predicates, function symbols and constants start with a lower-case letter.

The domain-independent definition of the holdsAt predicate is as follows:

$$
\begin{aligned}
\text{holdsAt}(\ F\!=\!V,\ T\ ) \leftarrow \\
\text{initially}(\ F\!=\!V\ ), \\
\text{not broken}(\ F\!=\!V,\ 0,\ T\ )
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
\text{holdsAt}(\ F\!=\!V,\ T\ ) \leftarrow \\
\text{happens}(\ Act,\ T'\ ), \\
T' < T, \\
\text{initiates}(\ Act,\ F\!=\!V,\ T'\ ), \\
\text{not broken}(\ F\!=\!V,\ T',\ T\ )
\end{aligned}
\tag{2}
$$

According to axiom (1) a fluent holds at time T if it held initially (time 0) and has not been 'broken' in the meantime, that is, terminated between times 0 and T. Axiom (2) specifies that a fluent holds at a time T if it was initiated at some earlier time T' and has not been terminated between T' and T. 'not' represents 'negation by failure' [3]. The domain-independent predicate broken is defined as follows:

$$
\begin{aligned}
\text{broken}(\ F\!=\!V,\ T_1,\ T_3\ ) \leftarrow \\
\text{happens}(\ Act,\ T_2\ ), \\
T_1 \leq T_2,\ T_2 < T_3, \\
\text{terminates}(\ Act,\ F\!=\!V,\ T_2\ )
\end{aligned}
\tag{3}
$$

$F\!=\!V$ is 'broken' between T_1 and T_3 if an event takes place in that interval that terminates $F\!=\!V$. A fluent cannot have more than one value at any time. The following domain-independent axiom captures this feature:

$$
\begin{aligned}
\text{terminates}(\ Act,\ F\!=\!V,\ T\ ) \leftarrow \\
\text{initiates}(\ Act,\ F\!=\!V',\ T\ ), \\
V \neq V'
\end{aligned}
\tag{4}
$$

Axiom (4) states that if an action Act initiates $F\!=\!V'$ then Act also terminates $F\!=\!V$, for all other possible values V of the fluent F. We do not insist that a fluent must have a value at every time-point. In this version of EC, therefore, there is a difference between initiating a Boolean fluent $F\!=\!\text{false}$ and terminating $F\!=\!\text{true}$: the first implies, but is not implied by, the second.

We make the following further comments regarding this version of EC. First, the domain-independent EC axioms (1)–(4) specify that a fluent does not hold at the time that was initiated but holds at the time it was terminated. Second, in addition to the presented domain-independent definitions, the holdsAt and terminates predicates may be defined in a domain-dependent manner. The happens, initially and initiates predicates are defined only in a domain-dependent manner. Third, in addition to axioms (1)–(4), the domain-independent axioms of EC include those defining the holdsFor predicate, that is, the predicate for computing the intervals in which a fluent holds. To save space we do not present here the definition of holdsFor; the interested

reader is referred to the source code of the long-term behaviour recognition (LTBR) system, which is available upon request.

3 Short-Term Behaviours: The CAVIAR Dataset

LTBR includes an EC action description expressing long-term behaviour definitions. The input to LTBR is a symbolic representation of short-term behaviours. The output of LTBR is a set of recognised long-term behaviours. In this paper we present experimental results given the short-term behaviours of the first dataset of the CAVIAR project[1]. This dataset includes 28 surveillance videos of a public space. The videos are staged — actors walk around, browse information displays, sit down, meet one another, leave objects behind, fight, and so on. Each video has been manually annotated in order to provide the ground truth for both short-term and long-term behaviours.

For this set of experiments the input to LTBR is: (i) the short-term behaviours walking, running, active and inactive, along with their time-stamps, that is, the frame in which a short-term behaviour took place, (ii) the coordinates of the tracked people and objects as pixel positions at each time-point, and (iii) the first time and the last time a person or object is tracked ('appears'/'disappears'). Given this input, LTBR recognises the following long-term behaviours: a person leaving an object, a person being immobile, people meeting, moving together, or fighting.

Short-term behaviours are represented as EC actions whereas the long-term behaviours that LTBR recognises are represented as EC fluents. In the following section we present example fragments of all long-term behaviour definitions.

4 Long-Term Behaviour Definitions

The long-term behaviour 'leaving an object' is defined as follows:

$$\text{initiates}(\ inactive(Object),\ leaving\_object(Person,\ Object) = \text{true},\ T\ ) \leftarrow$$
$$\qquad \text{holdsAt}(\ appearance(Object) = appear,\ T\ ),$$
$$\qquad \text{holdsAt}(\ close(Person,\ Object,\ 30) = \text{true},\ T\ ), \qquad\qquad (5)$$
$$\qquad \text{holdsAt}(\ appearance(Person) = appear,\ T_0\ ),$$
$$\qquad T_0 < T$$

$$\text{initiates}(\ exit(Object),\ leaving\_object(Person,\ Object) = \text{false},\ T\ ) \qquad (6)$$

Axiom (5) expresses the conditions in which a 'leaving an object' behaviour is recognised. The fluent recording this behaviour, *leaving_object*(*Person*, *Object*), becomes true at time T if *Object* is inactive at T, *Object* 'appears' at T, there is a

---

[1] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

Person close to *Object* at *T* (in a sense to be specified below), and *Person* has appeared at some time earlier than *T*. The *appearance* fluent records the times in which an object/person 'appears' and 'disappears'. The $close(A,B,D)$ fluent is true when the distance between *A* and *B* is at most *D*. The distance between two tracked objects/people is computed given their coordinates. Based on our empirical analysis the distance between a person leaving an object and the object is at most 30.

An object exhibits only inactive short-term behaviour. Any other type of short-term behaviour would imply that what is tracked is not an object. Therefore, the short-term behaviours active, walking and running do not initiate the *leaving_object* fluent. In the CAVIAR videos an object carried by a person is not tracked — only the person that carries it is tracked. The object will be tracked, that is, 'appear', if and only if the person leaves it somewhere. Consequently, given axiom (5), the *leaving_object* behaviour will be recognised only when a person leaves an object (see the second line of axiom (5)), not when a person carries an object.

Axiom (6) expresses the conditions in which a *leaving_object* behaviour ceases to be recognised. In brief, *leaving_object* is terminated when the object in question is picked up. $exit(A)$ is an event that takes place when $appearance(A) = disappear$. An object that is picked up by someone is no longer tracked — it 'disappears' — triggering an *exit* event which in turn terminates *leaving_object*.

The long-term behaviour *immobile* was defined in order to signify that a person is resting in a chair or on the floor, or has fallen on the floor (fainted, for example). Below is (a simplified version of) an axiom of the *immobile* definition:

$$
\begin{aligned}
\textsf{initiates}(\ &inactive(Person),\ immobile(Person) = \textsf{true},\ T\ ) \leftarrow \\
&\textsf{happens}(\ active(Person),\ T_0\ ), \\
&T_0 < T, \\
&duration(\ inactive(Person),\ Intervals\ ), \\
&(T,\ T_1)\ \in\ Intervals, \\
&T_1 > T + 54
\end{aligned}
\tag{7}
$$

According to axiom (7), the behaviour *immobile(Person)* is recognised if *Person*: (i) has been active some time in the past, and (ii) stays inactive for more than 54 frames (we chose this number of frames given our empirical analysis of the CAVIAR dataset). *duration* is a predicate computing the duration of inactive behaviour, that is, the number of consecutive instantaneous inactive events. The output of *duration* is a set of tuples of the form (s,e) where *s* is the time in which *inactive(Person)* started and *e* is the time in which *inactive(Person)* ended. Note that this is not the only way to represent durative events in EC. See [9] for alternative representations.

Axiom (7) has an additional constraint requiring that *Person* is not close to an information display or a shop — if *Person* was close to a shop then she would have to stay inactive much longer than 54 frames before *immobile* could be recognised. In this way we avoid classifying the behaviour of browsing a shop as *immobile*. To simplify the presentation we do not present here the extra constraint of axiom (7).

The definition of *immobile* includes axioms according to which *immobile(Person)* is recognised if *Person*: (i) has been walking some time in the past, and (ii) stays in-

active for more than 54 frames. We insist that *Person* in *immobile*(*Person*) has been active or walking before being inactive in order to distinguish between a left object, which is inactive from the first time it is tracked, from an immobile person.

immobile(*Person*) is terminated when *Person* starts walking, running or 'disappears' — see axioms (8)–(10) below:

$$\text{initiates}(\ walking(Person),\ immobile(Person) = \text{false},\ T\ ) \tag{8}$$

$$\text{initiates}(\ running(Person),\ immobile(Person) = \text{false},\ T\ ) \tag{9}$$

$$\text{initiates}(\ exit(Person),\ immobile(Person) = \text{false},\ T\ ) \tag{10}$$

The following axioms represent a fragment of the *moving* behaviour definition:

$$\begin{aligned}
\text{initiates}(\ & walking(Person),\ moving(Person,\ Person_2) = \text{true},\ T\ ) \leftarrow \\
& \text{holdsAt}(\ close(Person,\ Person_2, 34) = \text{true},\ T\ ), \\
& \text{happens}(\ walking(Person_2),\ T\ )
\end{aligned} \tag{11}$$

$$\begin{aligned}
\text{initiates}(\ & walking(Person),\ moving(Person,\ Person_2) = \text{false},\ T\ ) \leftarrow \\
& \text{holdsAt}(\ close(Person,\ Person_2,\ 34) = \text{false},\ T\ )
\end{aligned} \tag{12}$$

$$\begin{aligned}
\text{initiates}(\ & active(Person),\ moving(Person,\ Person_2) = \text{false},\ T\ ) \leftarrow \\
& \text{happens}(\ active(\ Person_2\ ),\ T\ )
\end{aligned} \tag{13}$$

$$\text{initiates}(\ running(Person),\ moving(Person,\ Person_2) = \text{false},\ T\ ) \tag{14}$$

$$\text{initiates}(\ exit(Person),\ moving(Person,\ Person_2) = \text{false},\ T\ ) \tag{15}$$

According to axiom (11) *moving* is initiated when two people are walking and are close to each other (their distance is at most 34). *moving* is terminated when the people walk away from each other, that is, their distance becomes greater than 34 (see axiom (12)), when they stop moving, that is, become active (see axiom (13)) or inactive, when one of them starts running (see axiom (14)), or when one of them 'disappears' (see axiom (15)).

The following axioms express the conditions in which *meeting* is recognised:

$$\begin{aligned}
\text{initiates}(\ & active(Person),\ meeting(Person,\ Person_2) = \text{true},\ T\ ) \leftarrow \\
& \text{holdsAt}(\ close(Person,\ Person_2,\ 25) = \text{true},\ T\ ), \\
& \text{not happens}(\ running(Person_2),\ T\ )
\end{aligned} \tag{16}$$

$$\begin{aligned}
\text{initiates}(\ & inactive(Person),\ meeting(Person,\ Person_2) = \text{true},\ T\ ) \leftarrow \\
& \text{holdsAt}(\ close(Person,\ Person_2,\ 25) = \text{true},\ T\ ), \\
& \text{not happens}(\ running(Person_2),\ T\ )
\end{aligned} \tag{17}$$

meeting is initiated when two people 'interact': at least one of them is active or inactive, the other is not running, and the distance between them is at most 25. This interaction phase can be seen as some form of greeting (for example, a handshake). *meeting* is terminated when the two people walk away from each other, or one of them starts running or 'disappears'. The axioms representing the termination of *meeting* are similar to axioms (12), (14) and (15). Note that *meeting* may overlap with *moving*: two people interact and then start *moving*, that is, walk while

being close to each other. In general, however, there is no fixed relationship between *meeting* and *moving*.

The axioms below present the conditions in which *fighting* is initiated:

$$\begin{aligned}
\text{initiates}(\ active(Person),\ fighting(Person,\ Person_2) = \text{true},\ T\ ) \leftarrow \\
\text{holdsAt}(\ close(Person,\ Person_2,\ 24) = \text{true},\ T\ ), \\
\text{not happens}(\ inactive(Person_2),\ T\ )
\end{aligned} \quad (18)$$

$$\begin{aligned}
\text{initiates}(\ running(Person),\ fighting(Person,\ Person_2) = \text{true},\ T\ ) \leftarrow \\
\text{holdsAt}(\ close(Person,\ Person_2,\ 24) = \text{true},\ T\ ), \\
\text{not happens}(\ inactive(Person_2),\ T\ )
\end{aligned} \quad (19)$$

Two people are assumed to be *fighting* if at least one of them is active or running, the other is not inactive, and the distance between them is at most 24. We have specified that running initiates *fighting* because, in the CAVIAR dataset, moving abruptly, which is what happens during a fight, is often classified as running. *fighting* is terminated when one of the people walks or runs away from the other, or 'disappears' — see axioms (20)–(22) below:

$$\begin{aligned}
\text{initiates}(\ walking(Person),\ fighting(Person,\ Person_2) = \text{false},\ T\ ) \leftarrow \\
\text{holdsAt}(\ close(Person,\ Person_2,\ 24) = \text{false},\ T\ )
\end{aligned} \quad (20)$$

$$\begin{aligned}
\text{initiates}(\ running(Person),\ fighting(Person,\ Person_2) = \text{false},\ T\ ) \leftarrow \\
\text{holdsAt}(\ close(Person,\ Person_2,\ 24) = \text{false},\ T\ )
\end{aligned} \quad (21)$$

$$\text{initiates}(\ exit(Person),\ fighting(Person,\ Person_2) = \text{false},\ T\ ) \quad (22)$$

Under certain circumstances LTBR recognises both *fighting* and *meeting* — this happens when two people are active and the distance between them is at most 24. This problem would be resolved if the CAVIAR dataset included a short-term behaviour for abrupt motion, which would be used (instead of the short-term behaviour active) to initiate *fighting*, but would not be used to initiate *meeting*.

5 Experimental Results

We present our experimental results on 28 surveillance videos of the CAVIAR project. These videos contain 26419 frames that have been manually annotated in order to provide the ground truth for short-term and long-term behaviours. Table 2 shows the performance of LTBR — we show, for each long-term behaviour, the number of True Positives (TP), False Positives (FP) and False Negatives (FN), as well as Recall and Precision.

LTBR correctly recognised 4 *leaving_object* behaviours. Moreover, there were no FP. On the other hand, there was 1 FN. This, however, cannot be attributed to LTBR because in the video in question the object was left behind a chair and was not tracked. In other words, the left object never 'appeared', it never exhibited a short-term behaviour.

Table 2 Experimental Results.

Behaviour	True Positive	False Positive	False Negative	Recall	Precision
leaving object	4	0	1	0.8	1
immobile	9	8	0	1	0.52
moving	16	12	1	0.94	0.57
meeting	6	3	3	0.66	0.66
fighting	4	8	2	0.66	0.33

Regarding *immobile* we had 9 TP, 8 FP and no FN. The recognition of *immobile* would be much more accurate if there was a short-term behaviour for the motion of leaning towards the floor or a chair. Due to the absence of such a short-term behaviour, the recognition of *immobile* is primarily based on how long a person is inactive. In the CAVIAR videos a person falling on the floor or resting in a chair stays inactive for at least 54 frames. Consequently LTBR recognises *immobile* if, among other things, a person stays inactive for at least 54 frames (we require that a person stays inactive for a longer time period if she is located close to a shop to avoid FP when a person is staying inactive browsing a shop). There are situations, however, in which a person stays inactive for more than 54 frames and has not fallen on the floor or sat in a chair: people watching a fight, or just staying inactive waiting for someone. It is in those situations that we have the FP concerning *immobile*. We expect that in longer videos recording actual behaviours (as opposed to the staged behaviours of the CAVIAR videos) a person falling on the floor or resting in a chair would be inactive longer than a person staying inactive while standing. In this case we could increase the threshold for the duration of inactive behaviour in the definition of *immobile*, thus potentially reducing the number of FP concerning *immobile*.

LTBR recognised correctly 16 *moving* behaviours. However, it also recognised incorrectly 12 such behaviours. Half of the FP concern people that do move together: walk towards the same direction while being close to each other. According to the manual annotation of the videos, however, these people do not exhibit the *moving* long-term behaviour. The remaining FP fall into two categories. First, people walk close to each other as they move to different directions — in such a case the duration of a FP is very short. We may eliminate these FP by adding a constraint that the duration of *moving* is greater than a specified threshold. Second, the short-term behaviours of people *fighting* are sometimes classified as walking. Consequently, the behaviour of these people is incorrectly recognised by LTBR as *moving* since, according to the manual annotation of the CAVIAR dataset, they are walking while being close to each other (moreover, their coordinates change). Introducing a short-term behaviour for abrupt motion would resolve this issue, as abrupt motion would not initiate *moving*.

LTBR did not recognise 1 *moving* behaviour. This FN was due to the fact that the distance between the people walking together was greater than the threshold we have specified. Increasing this threshold would result in substantially increasing the number of FP. Therefore we chose not to increase it.

LTBR recognised 9 *meeting* behaviours, 6 of which took place and 3 did not take place. 2 FP concerned *fighting* behaviours realised by people being active and close to each other. As mentioned in the previous section, in these cases LTBR recognises both *meeting* and *fighting*. The third FP was due to the fact that two people were active and close to each other, but were not interacting. LTBR did not recognise 3 *meeting* behaviours. 2 FN were due to the fact that the distance between the people in the meeting was greater than the threshold we have specified. If we increased that threshold LTBR would correctly recognise these 2 *meeting* behaviours. However, the number of FP for *meeting* would substantially increase. Therefore we chose not to increase the threshold distance. The third FN was due to the fact that the short-term behaviours of the people interacting — handshaking — were classified as walking (although one of them was actually active). We chose to specify that walking does not initiate a *meeting* in order to avoid incorrectly recognising meetings when people simply walk close to each other.

Regarding *fighting* we had 4 TP, 8 FP and 2 FN. The FP were mainly due to the fact that when a meeting takes place LTBR often recognises the long-term behaviour *fighting* (as well as *meeting*). LTBR did not recognise 2 *fighting* behaviours because in these two cases the short-term behaviours of the people *fighting* were classified as walking (recall the discussion on the recognition of *moving*). We chose to specify that walking does not initiate *fighting*. Allowing walking to initiate *fighting* (provided, of course, that two people are close to each other) would substantially increase the number of FP for *fighting*, because *fighting* would be recognised every time a person walked close to another person.

6 Discussion

We presented our approach to behaviour recognition. As demonstrated by the presented experiments, the use of EC allows for the development of a recognition system capable of dealing with, among other things, durative (short-term and long-term) behaviours, temporally overlapping, repetitive, and 'forbidden' behaviours, that is, behaviours that should not take place within a specified time-period in order to recognise some other behaviour (see [9, 8] for presentations of the expressiveness of EC). Furthermore, the availability of the full power of logic programming, which is one of the main attractions of employing EC as the temporal formalism, allows for the development of behaviour definitions including complex temporal and atemporal constraints. The majority of behaviour recognition systems (see [5, 4, 11, 10, 7] for a few well-known examples) employ less formal and less expressive formalisms to represent the properties of behaviours. For example, the well-known *chronicle recognition system*[2] (a 'chronicle' can be seen as a long-term behaviour) does not support any form of spatial reasoning and thus cannot be directly used for behaviour recognition in video surveillance applications. Our approach for more expressive be-

---

[2] `http://crs.elibel.tm.fr/`

haviour definitions leads to the recognition of more complex behaviours. Although the practicality of our approach, in terms of real-time recognition, remains to be investigated — this is an area of current research — the presented work can, at the very least, be applied to post-mortem analysis. A thorough comparison of our work with related research is not possible in the available space — such a comparison will be presented elsewhere.

We outline two directions for future research. First, we plan to perform behaviour recognition using datasets exhibiting a finer classification of short-term behaviours — for instance, explicitly representing abrupt motion. Second, we aim to employ inductive logic programming (ILP) techniques for fine-tuning in an automated way behaviour definitions (see, for example, [1] for an application of ILP techniques on EC formalisations).

References

1. D. Alrajeh, O. Ray, A. Russo, and S. Uchitel. Extracting requirements from scenarios with ILP. In *Inductive Logic Programming*, volume LNAI 4455. Springer, 2007.
2. A. Artikis, M. Sergot, and J. Pitt. Specifying norm-governed computational societies. *ACM Transactions on Computational Logic*, 10(1), 2009.
3. K. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Databases*, pages 293–322. Plenum Press, 1978.
4. C. Dousson and P. Le Maigat. Chronicle recognition improvement using temporal focusing and hierarchisation. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 324–329, 2007.
5. M. Ghallab. On chronicles: Representation, on-line recognition and learning. In *Proceedings of Conference on Principles of Knowledge Representation and Reasoning*, pages 597–606, 1996.
6. R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–96, 1986.
7. D. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, 2002.
8. R. Miller and M. Shanahan. The event calculus in a classical logic — alternative axiomatizations. *Journal of Electronic Transactions on Artificial Intelligence*, 4(16), 2000.
9. M. Shanahan. The event calculus explained. In M. Wooldridge and M. Veloso, editors, *Artificial Intelligence Today*, LNAI 1600, pages 409–430. Springer, 1999.
10. V.-T. Vu. *Temporal Scenarios for Automatic Video Interpretation*. PhD thesis, Université de Nice — Sophia Antipolis, 2004.
11. V.-T. Vu, F. Brémond, and M. Thonnat. Automatic video interpretation: A novel algorithm for temporal scenario recognition. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1295–1302, 2003.