CASE^E: A Hierarchical Event Representation for the Analysis of Videos

Asaad Hakeem, Yaser Sheikh, and Mubarak Shah
University of Central Florida, Orlando, FL 32816
{ahakeem, yaser, shah}@cs.ucf.edu

Abstract

A representational gap exists between low-level measurements (segmentation, object classification, tracking) and high-level understanding of video sequences. In this paper, we propose a novel representation of events in videos to bridge this gap, based on the CASE representation of natural languages. The proposed representation has three significant contributions over existing frameworks. First, we recognize the importance of causal and temporal relationships between sub-events and extend CASE to allow the representation of temporal structure and causality between sub-events. Second, in order to capture both multi-agent and multi-threaded events, we introduce a hierarchical CASE representation of events in terms of sub-events and case-lists. Last, for purposes of implementation we present the concept of a temporal event-tree, and pose the problem of event detection as subtree pattern matching. By extending CASE, a natural language representation, for the representation of events, the proposed work allows a plausible means of interface between users and the computer. We show two important applications of the proposed event representation for the automated annotation of standard meeting video sequences, and for event detection in extended videos of railroad crossings.

Introduction

Human community and society are built upon the ability to share experiences of events. Hence, in the enterprize of machine vision, the ability to represent and share observed events must be one of the ultimate, if most abstract, goals. With computer vision techniques maturing sufficiently to provide reliable low-level descriptions of scenes, the necessity of developing semantically meaningful descriptions of these low-level descriptors is becoming increasingly pressing. In this work, one primary objective is to present a coherent representation of events, as a means to encode the relationships between agents and objects participating in an event. We also emphasize, in particular, a representation that allows computers to share observations with other computers and also with humans, in terms of events. An *event* is defined as a collection of actions performed by one or more agents. *Agents* are animates that can perform actions independently or dependently (e.g. people or robots). The practical need for formal representation of events is best illustrated through possible applications. These applications include: (1) *Surveillance*: By definition, surveillance applications require the detection of peculiar events. Event representations can be used for prior definition of what constitutes an interesting event in any given domain, allowing automation of area surveillance, (2) *Video Indexing and Event Browsing*: Given a query for a certain event (defined in terms of an event representation), similar instances can be retrieved from a database of annotated clips, (3) *Annotation*: In the spirit of MPEG-7, video sequences may be annotated autonomously based on their content, (4) *Domain Understanding*: It is noted that causality is an abstract that cannot be directly inferred from a single video sequence. Through the use of event representations, causality can be inferred between events in a single domain (e.g. surveillance of airports) across several extended video sequences for domain understanding.

In literature, a variety of approaches have been proposed for the detection of events in video sequences. Most of these approaches can be arranged into two categories based on the semantic significance of their representations. This distinction is important, since it determines whether humans can exploit the representation for communication. Approaches where representations do not take on semantic meaning include Causal events (Brand 1997), Force dynamics (Siskind 2000), Stochastic Context Free Grammars (Bobick and Ivanov 1998), Spatio-temporal Derivatives (Zelnik-Manor and Irani 2001), and geometric properties and appearance (Malliot, Thonnat, and Boucher 2003). While they differ in approaches, the representations they employ do not lend themselves directly to interpretation or interface to humans. Learning methods such as Bayesian Networks and Hidden Markov Models (Ivanov and Bobick 2000) have been widely used in the area of activity recognition. A known drawback of learning methods is that they usually require large training sets of events and variation in data may require complete re-training. Similarly, there is no straightforward method of expanding the domain, once training has been completed. On the other hand, semantically significant approaches like the state machines (Koller, Heinze, and Nagel 1991), and PNF Networks (Pinhanez and Bobick 1998) provide varying degrees of representation to the actions and agents involved in the events.

What is missing in these representations is coherence in describing low-level measurements as 'events'. Can these representations be used to share knowledge between two

systems? Can events be compared on the basis of these representations? How are these representations related to human understanding of events? Can a human communicate his or her observation of an event to a computer or vice versa? By extending automatic generation of a natural language ontology to event representation of a video, a plausible interface between the human and the computer is facilitated. One such natural language representation called CASE was proposed by Fillmore (Fillmore 1968) for language understanding. The basic unit of this representation is a case-frame that has several elementary cases, such as an agentive, an instrumental, and a predicate. Using these case-frames Fillmore analyzed languages, treating *all* languages generically. However, CASE was primarily used for syntactic analysis of natural languages, and while it provides a promising foundation for event representation it has several limitations for that end. Firstly, since events are typically made up of a hierarchy of sub-events it is impossible to describe them as a succession of case-frames. Second, these sub-events often have temporal and causal relationships between them, and CASE provides no mechanisms to represent these relationships. Furthermore, there might be simultaneous dependent or independent sub-events with multiple agentives, and change of location and instrumentals during events. CASE was first investigated for event representation (Neumann 1989), but the author did not investigate the temporal structure of events as the author was not concerned with event *detection*. More recently (Kojima, Tamura, and Fukunaga 2001) addressed some shortcomings in CASE for single person event detection with, **SO**- (source prefixed to case), **GO**- (goal prefixed to case) and **SUB** (child frame describing a sub-event). **SO**- and **GO**- are prefixed to the **LOC** (locative) case mostly describing the source and destination locations of the agent in the event. A concept hierarchy of action rules (case-frames) was used to determine an action grammar (ontology) for the sequence of events. Also, using case-frames based on events, they reconstructed the event sequence in the form of sentences. Their method worked well for single person action analysis using the CASE representation. However, this work did not address important issues of temporal and causal relationships. Moreover, no mechanisms were proposed for multiple-agents or multi-threaded events.

We propose three critical extensions to CASE for the representation of events: (1) accommodating multiple agents and multi-threaded event, (2) supporting the inclusion of temporal information or *temporal logic* into the representation, and (3) supporting the inclusion of *causal* relationships between events as well. We also propose a novel event-tree representation, based on temporal relationships, for the detection of events in video sequences. Hence, unlike almost all previous work, we use both temporal structure and an environment descriptor simultaneously to represent an event.

The Extended CASE framework: CASE^E

In this section, the three extensions to the CASE framework are presented. Firstly, in order to capture both multi-agent and multi-thread events, we introduce a hierarchical CASE representation of events in terms of sub-events and case-lists. Secondly, since the temporal structure of events is critical to understanding and hence representing events, we introduce temporal logic into the CASE representation based on the interval algebra in (Allen and Ferguson 1994). Lastly, we recognize the importance of causal relationships between sub events and extend CASE to allow the representation of such causality between sub-events.

Multi-Agent, Multi-Thread Representation

Except in constrained domains, events typically involve multiple agents engaged in several dependent or independent actions. Thus any representation of events must be able to capture the composite nature of *real* events. To represent multiple objects, we introduce the idea of case-lists of elements for a particular case. For example, if there are more than one agents involved in an event we add both in a case-list within **AG**,

[**PRED**: move, **AG**:{ person1, person2 }, ...]

As in (Kojima, Tamura, and Fukunaga 2001), we use **SUB** to represent a sub-event that occurs during an event. However, this representation offers no means to represent *several* sub-events or multiple threads. To represent multiple threads we add them to a list of sub-events in the **SUB** case. An example is shown below,

"While Jack stole from the cashier, Bonnie robbed from the bank as Clyde was talking to the cashier"

[**PRED**: steal, **AG**: Jack, **D**: cashier, **SUB**:{
[**PRED**: rob, **AG**: Bonnie, **OBJ**: bank],
[**PRED**: talk, **AG**: { Clyde, cashier }] }
]

It should be immediately evident to the reader that the above event representation is ambiguous as temporal relations have not been defined. When did Clyde talk to the cashier? Was it before, during or after the steal event. In order to have an unambiguous representation, we need to incorporate temporal relations in our representation. The temporal relations are based on temporal logic, which is described in the next section.

Temporal Logic

The temporal structure of events is critical to understanding and hence representing events. Events are rarely instantaneous and often largely defined by the temporal order and relationship of their sub-events. In order to represent temporal relationships of sub-events, we introduce temporal logic into the CASE representation based on the interval algebra of (Allen and Ferguson 1994). We use this algebra to represent seven temporal relationships[1]. This interval temporal logic is shown in Fig. 1. Since temporal relationship exist between sub-events, the temporal case is always used in conjunction with **SUB**. The temporal case (e.g. **AFTER**) assumes the value of the predicate of the case-frame with which the temporal relationship exists. Consider, once again, the above example of the bank robbery by Bonnie and

[1] A minor modification was made where **BEFORE** was replaced by **AFTER** with respective modification in the parameters for ease of use in CASE

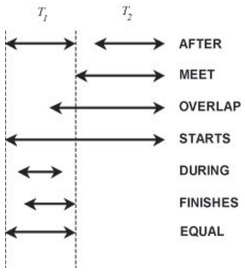


Figure 1: Allen's interval algebra describing temporal logic between durations T_1 and T_2.

Clyde, the case-frames with the temporal logic incorporated are,

[**PRED**: steal, **AG**: Jack, **D**: cashier, **SUB**:{
[**PRED**: rob, **AG**: Bonnie, **OBJ**: bank, **DURING**: steal],
[**PRED**: talk, **AG**: { Clyde, cashier }, **OVERLAP**: steal] }
]

The entire list of temporal cases, for two durations T_1 and T_2 is as follows,

AFTER : $T_2^{start} > T_1^{end}$
MEETS : $T_1^{end} = T_2^{start}$
DURING : $(T_1^{start} < T_2^{start}) \wedge (T_1^{end} > T_2^{end})$
FINISHES : $(T_1^{end} = T_2^{end}) \wedge (T_1^{start} < T_2^{start})$
OVERLAPS : $(T_1^{start} < T_2^{start}) \wedge (T_1^{end} > T_2^{start}) \wedge (T_1^{end} < T_2^{end})$
EQUAL : $(T_1^{start} = T_2^{start}) \wedge (T_1^{end} = T_2^{end})$
STARTS : $(T_1^{start} = T_2^{start}) \wedge (T_1^{end} \neq T_2^{end})$

It is ensured that each temporal case is unique. A little thought should convince the reader that temporal relationships between more than two events are also possible within this scheme.

There is still a requirement for representing the *dependency* of events. Some events require a causal relationship, i.e. they will not occur independently and are conditional upon other events. The representation so far does not have the capability to codify causal relationships, which is addressed in the next section.

Causality

In understanding the nature of events, the causal relationships between the constituent sub-events are indispensable. Some events might not occur if certain conditions were not satisfied, while some events may be dependent on other events. In order to explain this concept we show a simplistic example below,

"Caravaggio pulled the chair therefore Michelangelo fell down."

[**PRED**: pull, **AG**: Caravaggio, **OBJ**: chair, **CAUSE**:
[**PRED**: fall, **D**: Michelangelo, **FAC**: down]]

In the above example, Michelangelo would not have fallen down if Caravaggio had not pulled the chair. Therefore the 'fall' and 'pull' event have a causal relationship. It should be noted that only definite causal relations are represented by the **CAUSE** case, instead of using **SUB**. While the proposed extension allows the representation of causal relationships, it is noted that causal relationships cannot be inferred from video measurements alone. In other words, it is impossible to make a distinction between two successive events, and two causal events without some reasoning. Thus, from the point of view of on-line processing of measurements, videos are represented in terms of a *temporal representation*. Events and sub-events are arranged in a hierarchy according to the order of their temporal incidence and duration. Inferring causality solely from these temporal representations is a promising future direction.

Event Detection in Videos

In this section, we address some issues of implementing the proposed representation for event detection in videos. Video data is available as a discrete set of images, sampled on sequential lattices. Let $f(\mathbf{p}, t)$ represent a continuous video signal, indexed by spatial and temporal coordinates respectively. By indexing on the discrete-time variable k we can *temporally* represent the video signal as the set $\{f[\mathbf{x}, k]\}$ for $1 \leq k \leq N$, where N is the temporal support of the video signal, and $\mathbf{x} = (x, y)$ denotes the spatial coordinate (over some support). Here it is assumed that the lower-level tasks of object detection, classification and tracking have been performed for a stationary camera (corresponding to the GSD of Neumann (1989)). Each object is represented in terms of its label and motion, e.g. $\{\mathsf{person}_a,\ \mathbf{u}_a\}$, where $\mathbf{u}_a = \{\ (x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)\ \}$ is the trajectory of person_a's centroid. It is important to note that since it is the *relative* concept of motion that we are interested in (e.g. where did agent_1 move to with respect to object_2?), two-dimensional projections of three-dimensional world trajectories are sufficient for event representation (barring some degenerate configurations). Particular to each domain, domain objects and a vocabulary of predicates can be defined. Conceptually, these are the 'input' into a system that represents these inputs in terms of CASE^E. For event *detection*, a set of events are predefined as events of interest. In order to detect these events of interest within an autonomously generated representation of events in a video sequences, we pose the problem as a subtree isomorphism. A similarity measure is defined to guide the search for a match.

Maximal Subtree Isomorphism

The temporal structure of CASE^E can be intuitively visualized as a rooted tree, with each vertex corresponding to a sub-event (case-frame), and each edge corresponding to the temporal relationship between two vertices (e.g. **AFTER**, **MEET**). A split occurs at the simultaneous incident of multiple sub-events or when one of several sub-event ends during a parent event. An example sub-tree is shown in Figure 2. The problem of detecting the occurrence of a pre-defined event can be posed as finding a maximal subtree isomorphism. Given a video stream, a rooted tree can be continuously grown based on temporal relations of sub-events. Each pre-defined event itself can be represented as a tree, too. For two rooted trees, $T_1 = (V_1, E_1)$, the event-tree of the pre-defined events of

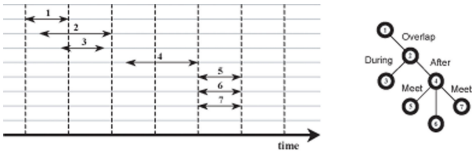


Figure 2: An Event Tree. The plot shows the temporal duration of a succession of sub-events, and to the right of it is a tree representation. Each vertex corresponds to a sub-event, and each edge represents the temporal relationship with the parent vertex.

interest, and $T_2 = (V_2, E_2)$, the video event-tree, any bijection $\phi : H_1 \rightarrow H_2$, with $H_1 \subseteq V_1$ and $H_2 \subseteq V_2$, is a subtree isomorphism if it preserves the adjacency and the hierarchy between vertices and if $T_1(H_1)$ and $T_2(H_2)$ are trees too. In our implementation, we employ a naive search for the maximal subtree isomorphism. A more sophisticated algorithm has been proposed in (Pelillo, Siddiqi, and Zucker 1999), where it shown that there is a one-to-one correspondence between the maximal subtree isomorphism problem and the maximal clique problem and, moreover, that a continuous formulation of the maximum clique problem can be exploited to find a solution to the original subtree matching problem. The cost function we wish to maximize is the similarity of corresponding vertices, and the number of matching edges. To that end, a measure of similarity between case-frames, which correspond to vertices in the event-tree, is defined next.

Similarity Most pattern recognition problems are based on feature vectors of real-valued numbers usually with a natural measure of distance between vectors. On the other hand, in measuring the 'similarity' between nominal data as in our case there is no clear notion of similarity (or metric). However, within a *complete* event domain, where a complete event domain is one where the set of possible objects and vocabulary is *uniquely* defined, it is possible to measure similarity between case-frames. We wish to compare a pair of observations $(\mathbf{C}_1, \mathbf{C}_2)$, where $\mathbf{C}_i = [c_{i1}, c_{i2}, \ldots, c_{ip}]$ corresponds to a case-frame and each element corresponds to a case. If a certain case does not exist (e.g. if the location is unspecified) the value of the element is Ø. Now for a complete event domain, we can define $\psi(c_{ik}, c_{jk})$,

$$\psi(c_{ik}, c_{jk}) = \begin{cases} \infty & \text{if } c_{ik} = c_{jk} = \emptyset \\ 1 & \text{if } c_{ik} = c_{jk} \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

and similarity is measured using the Jaccard coefficient (ratio of sizes of intersection and union),

$$\rho(\mathbf{C_i}, \mathbf{C_j}) = \frac{\sum_{k=1}^{p} \mathbf{I}(\psi(c_{ik}, c_{jk}) = 1)}{\sum_{k=1}^{p} \mathbf{I}(\psi(c_{ik}, c_{jk}) = 1) + \sum_{k=1}^{p} \mathbf{I}(\psi(c_{ik}, c_{jk}) = 0)} \quad (1)$$

where $\mathbf{I}$ is an indicator function. An evaluation of the Jaccard Coefficient is shown in Figure 3.

Experiments and Discussion

We performed two sets of experiments (corresponding to each domain), both were implemented to run in real time (30 fps) on a 2.1 GHz Pentium Machine. The first experiment set involved the standard PETS test video for hand posture classification, as well as 11 other unconstrained sequences of human interaction. Initial object identification and labelling were performed manually, and further tracking was performed using the MEANSHIFT tracking algorithm. The objective was to perform a real-time generation of CASE^E representations. Figures 4 shows snapshots of individuals interacting in an unconstrained environment and their corresponding event representations.

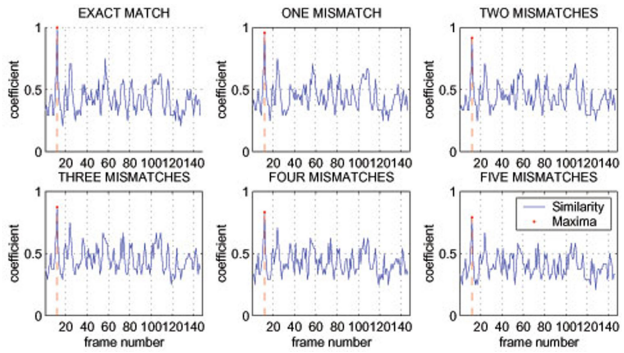
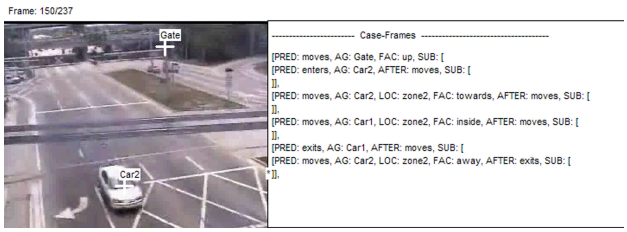
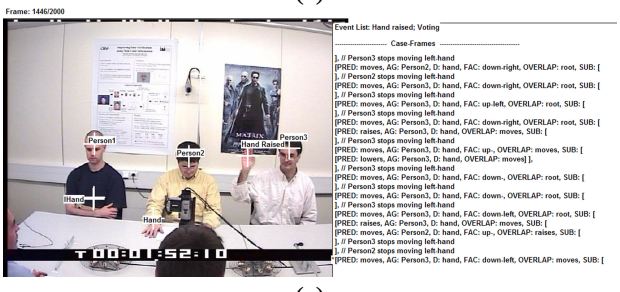


Figure 3: Matching using the Jaccard Coefficient. A predefined event-tree consisting of six vertices (case-frames) is matched with an event-tree of a video sequence consisting of 148 vertices (case-frames). The correct match occurs at the subtree rooted at frame 12 (the similarity maximum by the dotted red line). From top-left to bottom-right the pre-defined predicate is perturbed so that progressively greater number of case-elements within the case-frames mismatch.



(a)



(c)

Figure 4: On-line CASE^E representation of a video sequences. Frames during the online CASE^E generation are shown, with each agent labelled. (a) Representation at Frame 150/237 for the Railroad Monitoring Experiment (b) (PETS Sequence) Representation at Frame 1446/2000.

The second set of experiments was to the application of CASE^E to the domain of monitoring a railroad intersection. In an independent railway study of 25,000 video sequences, ten events of interest were identified in this domain. We encoded each of these events using the CASE^E representation and include CASE^E representations of these ten events. The encoding of human interaction domain using CASE^E is not shown due to space limitation, but CASE^E can represent events of two very different domains. The railroad scenario was selected since it is a closed environment suited to formal representation. For this domain, the CASE^E was coded for events of interest, and as the system produced CASE^E representations for the video stream, it monitored for the incidence of each pre-defined event using the algorithm previously presented. At the instance of each frame, the motion of each agent is analyzed and used to update the current CASE^E. Results of the event detection are shown in Figure 5. The experimental results for the precision and recall values are summarized in Table 1.

No. of Frames	Events Detected	Ground Truth	False Positive	Precision %	Recall %
272	18	17	1	94.44	100
311	24	27	2	91.67	81.48
330	40	36	5	87.50	97.22
161	18	16	3	83.33	93.87
187	39	32	7	82.05	100
184	13	13	1	92.03	92.03
165	18	18	0	100	100
247	40	38	4	90	94.73
342	61	51	11	81.96	98.03
2000	102	108	7	93.13	87.96
335	32	29	5	84.37	93.10
402	34	28	7	79.41	96.42
237	9	10	1	88.89	80
223	4	4	0	100	100
486	12	9	3	75	100
192	9	8	1	88.89	100

Table 1: Summary of results for the two experiment sets.

Application of CASE^E Representation for Monitoring Railroad Intersections

Domain Entities

Vehicle	A vehicle in the universe
Person	A person in the universe
Train	A train on the tracks
Gate	Gate at the railroad crossing
Signal	Signal at the railroad crossing
Zone1	Zone covering the area of activation for the signal
Zone2	Zone covering a designated high-risk area
Tracks	The tracks that the train travels on

Domain Predicates *Moves*, *Enters*, *Exits*, *Switches*, *Signals*, *Breaks*, *Collides*, *Stops*.

Domain Events

1. **train approaches ⇒ signal switches on ⇒ gate arm moves down ⇒ vehicle stops outside Zone2**
   [**PRED**: Moves, **AG**: Train, **D**: Signals, **LOC**: Zone1, **FAC**: Towards, **CAUSE**:
   [**PRED**: Switches, **AG**: Signals, **FAC**: On, **CAUSE**:
   [**PRED**: Moves, **AG**: Gate, **FAC**: Down, **AFTER**: Switches, **SUB**:
   [**PRED**: Stops, **AG**: Vehicle, **LOC**: Zone2, **FAC**: Outside, **AFTER**: Moves]]]]

2. **train approaches ⇒ signal switches on ⇒ gate arm moves down ⇒ vehicle stops inside Zone2**
   [**PRED**: Moves, **AG**: Train, **D**: Signals, **LOC**: Zone1, **FAC**: Towards, **CAUSE**:
   [**PRED**: Switches, **AG**: Signals, **FAC**: On, **CAUSE**:
   [**PRED**: Moves, **AG**: Gate, **FAC**: Down, **AFTER**: Switches, **SUB**:
   [**PRED**: Stops, **AG**: Vehicle, **LOC**: Zone2, **FAC**: Inside, **AFTER**: Moves]]]]

3. **train approaches ⇒ signal switches on ⇒ gate arm moves down ⇒ vehicle breaks the gate arm while entering Zone2**
   [**PRED**: Moves, **AG**: Train, **D**: Signals, **LOC**: Zone1, **FAC**: Towards, **CAUSE**:
   [**PRED**: Switches, **AG**: Signals, **FAC**: On, **CAUSE**:
   [**PRED**: Moves, **AG**: Gate, **FAC**: Down, **AFTER**: Switches, **SUB**:
   [**PRED**: Enters, **AG**: Vehicle, **LOC**: Zone2, **DURING**: Moves, **SUB**:
   [**PRED**: Breaks, **AG**: Vehicle, **D**: Gate, **DURING**: Enters]]]]]

4. **train approaches ⇒ signal switches on ⇒ gate arm moves down ⇒ vehicle breaks the gate arm while exiting Zone2**
   [**PRED**: Moves, **AG**: Train, **D**: Signals, **LOC**: Zone1, **FAC**: Towards, **CAUSE**:
   [**PRED**: Switches, **AG**: Signals, **FAC**: On, **CAUSE**:
   [**PRED**: Moves, **AG**: Gate, **FAC**: Down, **AFTER**: Switches, **SUB**:
   [**PRED**: Exits, **AG**: Vehicle, **LOC**: Zone2, **DURING**: Moves, **SUB**:
   [**PRED**: Breaks, **AG**: Vehicle, **D**: Gate, **DURING**: Exits]]]]]

5. **train approaches ⇒ signal switches on ⇒ gate arm moves down ⇒ vehicle enters while gate is in motion**
   [**PRED**: Moves, **AG**: Train, **D**: Signals, **LOC**: Zone1, **FAC**: Towards, **CAUSE**:
   [**PRED**: Switches, **AG**: Signals, **FAC**: On, **CAUSE**:
   [**PRED**: Moves, **AG**: Gate, **FAC**: Down, **AFTER**: Switches, **SUB**:
   [**PRED**: Enters, **AG**: Vehicle, **LOC**: Zone2, **DURING**: Moves]]]]

6. **train approaches ⇒ signal switches on ⇒ gate arm moves down ⇒ vehicle exits while gate is in motion**
   [**PRED**: Moves, **AG**: Train, **D**: Signals, **LOC**: Zone1, **FAC**: Towards, **CAUSE**:
   [**PRED**: Switches, **AG**: Signals, **FAC**: On, **CAUSE**:
   [**PRED**: Moves, **AG**: Gate, **FAC**: Down, **AFTER**: Switches, **SUB**:
   [**PRED**: Exits, **AG**: Vehicle, **D**: Zone2, **DURING**: Moves]]]]

7. **Vehicle collides with train**
   [**PRED**: Moves, **AG**: Train, **LOC**: Zone2, **FAC**: Inside, **SUB**:
   [**PRED**: Moves, **AG**: Vehicle, **FAC**: Inside, **LOC**: Zone2, **DURING**: Move, **CAUSE**:
   [**PRED**: Collides, **AG**: { Vehicle, Train }]]]

8. **Person being hit by train**
   [**PRED**: Moves, **AG**: Train, **LOC**: Zone2, **FAC**: Inside, **SUB**:
   [**PRED**: Moves, **AG**: Person, **FAC**: Inside, **LOC**: Zone2, **DURING**: Move, **CAUSE**:
   [**PRED**: Collides, **AG**: { Person, Train }]]]

9. **Person enters zone2 while signal was switched on**
   [**PRED**: Switches, **AG**: Signals, **FAC**: On, **SUB**:
   [**PRED**: Enters, **AG**: Person, **LOC**: Zone2, **DURING**: Switches]]

10. **Train entering zone2 while gates are in motion**
    [**PRED**: Moves, **AG**: Gates, **FAC**: Down, **SUB**:
    [**PRED**: Enters, **AG**: Train, **LOC**: Zone2, **DURING**: Moves]]

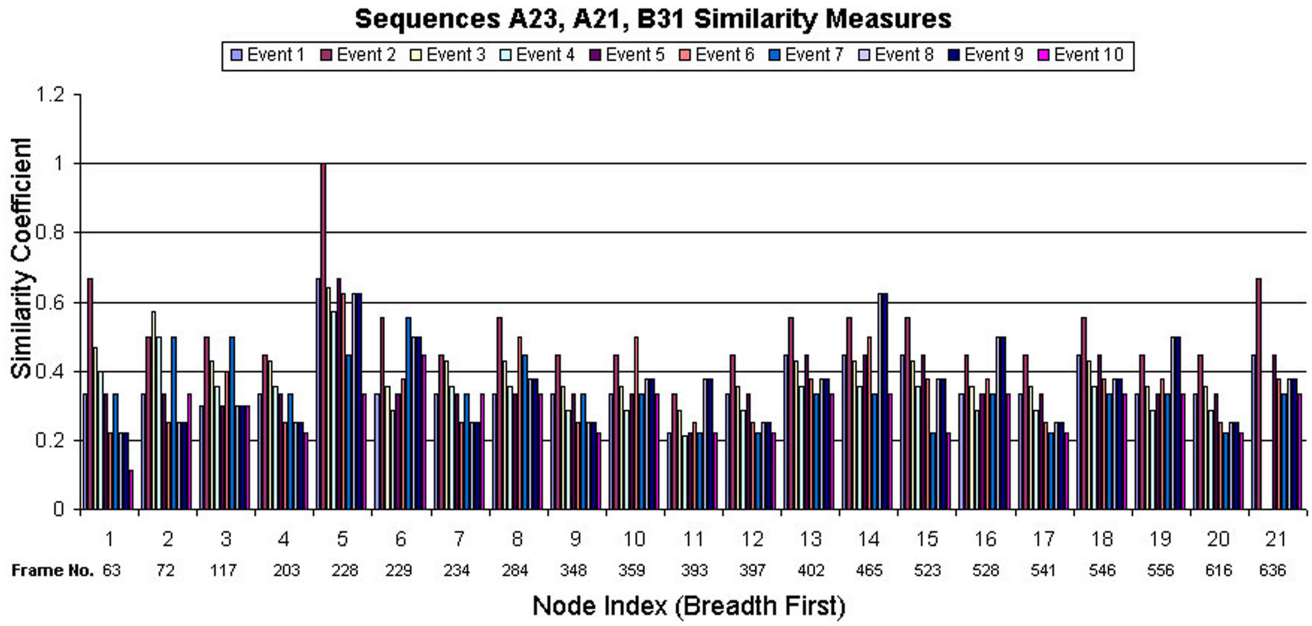


Figure 5: Event Detection using the Jaccard Coefficient. Sequences A23 (223 frames), A21 (237 frames), and B31 (192 frames) monitors railroad crossings (similar to the one shown in Figure 4). All events coded in the Application of CASE, we tested on the autonomously generated case frames and evidently Event 2 occurred at the time instant corresponding to Node Index 5 in sequence A21.

Conclusion

The problem of formally representing events occurring in a video sequence using measurements in terms of object labels and tracks was identified, and in order to represent events, cases were added to the original framework of (Fillmore 1968) to support multi-agent/thread, temporal logic and causal relationships. Experiments were performed on real sequences, for the on-line generation of CASE^E for human interaction, and a similarity coefficient was defined for the detection of pre-defined events. An instance of a complete event domain (Monitoring of Railroad intersections) was also treated. The essence of the proposition here is that based on the temporal relationships of the agent motions and a description of its state, it is possible to build a formal description of an event. We are interested in several future directions of this work including the inference of causality in video sequences, event-based retrieval of video, and unsupervised learning of event ontologies.

References

Allen, J. F., Ferguson, G. 1994. Actions and Events in Interval Temporal Logic. In *Journal of Logic and Computation*, Vol.4(5), pp.531-579.

Bobick, A. F., and Ivanov, Y. A. 1998. Action Recognition using Probabilistic Parsing. In *Proc. of Computer Vision and Pattern Recognition*, pp.196-202.

Brand, M. 1997. Understanding Manipulation in Video. In *International Conference on Face and Gesture Recognition*, pp.94-99.

Fillmore, C. J. 1968. The Case for CASE. In Bach, E. and Harms, R. eds., *Universals in Linguistic Theory*, pp.1-88, New York, NY:Holt, Rinehart, and Winston.

Ivanov Y. A., and Bobick A. F. 2000. Recognition of Visual Activities and Interactions by Stochastic Parsing. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.22, pp.852-872.

Kojima, A., Tamura, T. and Fukunaga, K. 2001. Natural Language Description of Human Activities from Video Images Based on Concept Hierarchy Actions. In *International Journal of Computer Vision*, vol.50, pp.171-184.

Koller, D., Heinze, N., and Nagel, H. H. 1991. Algorithmic Characterization of Vehicle Trajectories from Image Sequences by Motion Verbs. In *Proc. of Computer Vision and Pattern Recognition*, pp.90-95.

Maillot, N., Thonnat, M., and Boucher, A. 2003. Towards Ontology Based Cognitive Vision. In *Proc. of International Conference of Vision Systems*, pp.44-53.

Neumann, B. 1989. Natural Language Description of Time Varying Scenes. In Waltz, D. eds., *Semantic Structures: Advances in Natural Language Processing*, pp.167-206, Hillsdale, NJ: Lawrence Erlbaum Associates.

Pelillo, M., Siddiqi, K., and Zucker, S. 1999. Matching Hierarchical Structures using Association Graphs. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.21(11), pp.1105-1119.

Pinhanez, C., and Bobick, A. 1998. Human Action Detection Using PNF Propagation of Temporal Constraints. In *Proc. of Computer Vision and Pattern Recognition*, pp.898-904.

Siskind, J. M., 2000. Visual Event Classification via Force Dynamics. In *Proc. of Seventeenth National Conference on Artificial Intelligence*, pp.149-155, Menlo Park, CA:AAAI Press.

Zelnik-Manor, L., and Irani, M. 2001. Event-based Analysis of Video. *Proc. of Computer Vision and Pattern Recognition*, pp.123-130.