Abnormal Event Recognition: A Hybrid Approach Using Semantic Web Technologies

Luca Greco, Pierluigi Ritrovato, Alessia Saggese, Mario Vento
Dept. of Information Eng., Electrical Eng. and Applied Mathematics (DIEM), University of Salerno
Via Giovanni Paolo II, 132, 84084 Fisciano (SA) - Italy
{lgreco,pritrovato,asaggese,mvento}@unisa.it

Abstract

Video surveillance systems generated about 65% of the Universe Big Data in 2015. The development of systems for intelligent analysis of such a large amount of data is among the most investigated topics in the academia and commercial world. Recent outcomes in knowledge management and computational intelligence demonstrate the effectiveness of semantic technologies in several fields like image and text analysis, hand writing and speech recognition. In this paper a solution that, starting from the output of a people tracking algorithm, is able to recognize simple events (person falling to the ground) and complex ones (person aggression) is presented. The proposed solution uses semantic web technologies for automatically annotating the output produced by the tracking algorithm; a sets of rules for reasoning on these annotated data are also proposed. Such rules allow to define complex analytics functions demonstrating the effectiveness of hybrid approaches for event recognition.

1. Introduction and motivations

Video surveillance systems generated about 65% of the Universe Big Data in 2015 [1]. In recent years, more and more IP cameras have appeared around us for several purposes, mostly for security and monitoring activities. However, manual analysis of captured videos is a very time consuming operation and is typically not so accurate: indeed, it has been shown that, due to the psychological overcharge issue, *after 12 minutes of continuous video monitoring, a person will typically miss up to* 45% *of screen activity, while, after 22 minutes, up to* 95% *is overlooked.* Thus, a strong interest of the scientific community has been devoted to the development of automatic video analysis systems to detect events of interest and support the human operator in charge of monitoring tasks [14, 10].

A video analysis system works essentially through two main phases: (1) *detection and tracking*: objects moving in the scene are detected and tracked so as to extract the corresponding trajectories; (2) *events recognition*: trajectories are analyzed and eventually combined with context-based information so as to recognize events of interest. It is evident that the performance of the last phase is strongly dependent on the previous one: the more precise are the trajectories extracted during the first step, the better will be the performance of events' recognition system.

In the last fifteen years several tracking algorithms have been proposed in the literature [15, 2]. A common methodology is to describe objects moving within the scene using bottom up approaches: from raw pixels to objects, without any explicit knowledge about the context. Although achieved results are very promising, a definitive solution has not been found yet and several problems are still open: *missed objects*, related to persons (or in general objects of interest) present in the scene but not detected and tracked by the algorithm; *false positives*, namely spurious objects tracked by the system but not corresponding to any object of interest (and due, for instance, to moving trees); *id switches*, related to the fact that the trajectory of an object of interest is wrongly broken in two different tracks, having two different identifiers [7].

A knowledge driven approach can help to solve most of the above mentioned problems. Think, as an example, to spurious objects suddenly appearing in the scene, which typically cause false positives. Knowledge based systems may explicitly encode information related to entering areas (such as the borders of the scene as well as some doors located in the middle of the scene), so as to identify false positives as those objects appearing outside such areas. More generally, the following sources of knowledge can be exploited: (1) knowledge concerning the observed scene: entering or exiting areas; presence of particular objects in the scene which may cause false positive such as moving trees; occlusion areas, such as a pole or a wall; (2) knowledge con-

[1] T. Huang, Surveillance Video: The Biggest Big Data, Computing Now, vol. 7, no. 2, Feb. 2014, IEEE Computer Society [online]; http://www.computer.org/portal/web/computingnow/archive/february2014

cerning the particular typology of objects present inside the scene; (3) knowledge about the possible interactions among two or more objects moving in the scene.

Starting from the considerations above, in this paper we propose an hybrid approach that combines a traditional bottom-up approach with a top-down knowledge-driven one, where the contextual knowledge helps improving traditional pixel-based decisions. It is worth remembering that hybrid approaches have been successfully applied some years ago in different application fields, such as character or hand writing recognition: indeed, for several years the scientific community focused on the definition of the best possible combination of features to properly represent characters and train a classifier (a bottom-up data driven approach). However, the breakthrough in the performance of these systems was due to the introduction of context, that is the word to which a character belongs. It is also possible to note that a combination of bottom-up and top-down approaches is also exploited by the human brain when performing visual recognition tasks: for instance, it has been shown [11] that often the human being needs less time to recognize a word than a single letter belonging to that word. This apparently strange behavior of the brain can be justified by the fact that it starts trying to recognize the letter using fast but not very reliable features. If the combination of the letters forms a valid word (according to a dictionary), then the recognition task is completed.

Knowledge can be encoded in a machine understandable form in several ways. According to recent outcomes in the semantic web field, we propose to use the ontology to describe a specific domain. Given an ontology and a set of rules, any semantic web reasoner may be exploited in order to improve the decisions taken by traditional bottom up approaches.

This is why ontology and semantic web based approaches have been recently exploited for both tracking and activities recognition steps. The method proposed by [5] is one of the first examples in this context: semantic web is used to both obtain a high-level interpretation of the scene and improve the performance of traditional tracking procedures by explicitly represent the knowledge about the scene. However, one of the main limitations concerns the use of both standard (deductive) and non-standard (abductive) ontology-based reasoning, namely nRQL (new RACER Query Language) and RACER reasoner. It implies that interoperability with traditional web semantic methodologies becomes more difficult to achieve. In [13] knowledge about the scene is used with a different aim: semantic support and reasoning are only used for determining the best procedures, algorithms and system reactions to be applied during video analysis.

In [4] the authors focus on event recognition, especially for detecting tailgating from surveillance video. In particular, they propose one of the first ontology framework for this particular purpose, namely the Video Event Representation Language (VERL) and Video Event Markup Language (VEML). Unfortunately reasoning on VERL is computationally intensive and in some cases undecidable. In [12] the authors exploit OWL 2 for modeling and reasoning with complex human activities, while in [9] the authors propose a novel framework, SP-ACT, based on the combination of OWL and SPARQL, that is able to handle temporal relations between activities. The solutions proposed in these two papers demonstrate how semantic technologies can be used for describing activities and doing some inferences.

In this paper we propose an hybrid approach based on semantic technologies with a double aim: first, we demonstrate how to recognize and properly manage the typical issues of traditional bottom up tracking approaches. Second, we exploit a knowledge driven methodology based on the semantic web standards, namely OWL, SPARQL and SPIN, for reasoning on high level information, recognizing complex events and quickly developing advanced video analytics functionalities.

The paper is structured as follows. Section 1.1 introduces the semantic web technology stack. The proposed solution and how it works is described in Section 2 where the tracking ontology for the event recognition and examples of reasoning rules are presented. Finally, after presenting in Section 3 preliminary results, we draw some conclusions in Section 4.

1.1. Semantic Web Technology Stack

The Semantic Web has been described the first time in 2001 by Tim Berners-Lee, James Hendler and Ora Lassila in their famous paper *"The Semantic Web"* [1] as the evolution of the World Wide Web where the data available on the web are enriched with a meaning so that can be processed directly and indirectly by machines. The authors also argue *"... for the semantic web to function, computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning."*. The structured collection of information is defined by means of ontology: an explicit and formal specification of a conceptualization of a specific domain [6]. Therefore, ontologies are used for semantically annotate the data.

In order to achieve this vision, the World Wide Web Consortium has defined the Semantic Web technology stack depicted in Figure 1.

After 15 years from its presentation the Semantic Web is still under construction, but nowadays there exist concrete implementations. Examples are the Google knowledge graph, the Open Linked Data [2] just to mention few,

[2]Tim Berners-Lee (2006-07-27) "Linked Data Design Issues" W3C. http://www.w3.org/DesignIssues/LinkedData.html

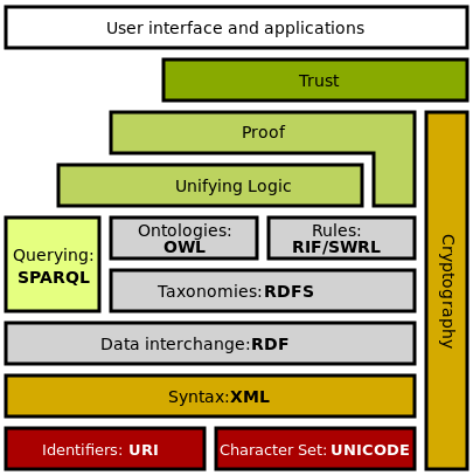

Figure 1. W3C Semantic Web technology Stack

both leveraging on mechanisms like schema.org[3], RDFa[4] and Microdata[5]. The semantic data annotation starts with the definition of an ontology schema. The current standard for Ontology definition and instantiation is Web Ontology Language (OWL) [8] which is defined as a family of markup languages based on the Resource Description Framework (RDF). RDF and OWL use Uniform Resource Identifiers (URI) to uniquely identify ontology elements (Figure 1). An ontology is composed by a series of axioms that assign restrictions to sets of individuals and relationships between individuals. Axioms are represented in terms of triple: *Subject, Predicate, Object*. If we consider `Frame`, `Camera` and `Blob` as three class of our tracking ontology, examples of triple are:

```
frame123 rdf:type Frame # the frame123 is
    an instance of the class Frame
camera1 rdf:type Camera
frame123 recordedBy camera1 # frame123 and
    camera1 are linked by the property
    recordedBy
blob321 belongTo frame123 .
```

The triples can be stored in RDF Triple stores (semantic database systems) and searched using the SPARQL query language (SPARQL Protocol and RDF Query Language). Ontology axioms can be analyzed by *inference engines* which infer new information from explicitly asserted data using a deductive process called *Reasoning*: one or more logical premises bring to a specific conclusion. Conversely, rule-based reasoning acts on the semantic knowledge by applying one or more predefined rules to add a new information. The problem with ontology based reasoners is that they support only deductive reasoning, *i.e.* simple IF *something* THEN *consequence* statements that express certainty in a sequence of events. However, scene interpretation needs abductive reasoning [5], *i.e.* taking a set of facts as input and finding a suitable hypothesis that explains them. Considering this, our methodology follows one of the most recent rule language specification, the W3C member submission SPARQL Inferencing Notation (SPIN) [6]. SPIN is a SPARQL based rule language. SPIN rules correspond to SPARQL queries which can be used to assert new facts, create new individuals or compute the probability of a certain event (abductive reasoning). The great advantage of using SPIN stays also in the possibility to use, in addition to SPARQL, other languages for defining a rule like Javascript and JAVA.

2. The proposed approach

2.1. Hybrid solution system overview

The architecture of the proposed hybrid system is depicted in Figure 2. As for the tracking component, we make use of one proposed by Foggia et al. in [3]. The selected tracking component uses a background subtraction algorithm for the detection phase and a state finite automata for tracking simultaneously people and groups of people. It is important to highlight that we properly set-up the configuration parameters so as to increase the sensitivity of the tracking algorithms by minimizing the number of misses, even increasing the number of false positive. Indeed, the knowledge component tries to give a meaning to the tracking output and is able to properly manage both ghost and noise objects, but cannot recover misses objects.

For each frame, the tracking component provides the ID of the identified object (in terms of persons and groups) together with the top-left and bottom-right vertex coordinates and the colors histogram for the upper-part and lower-part of the detected blob. The algorithm provides also the ID of persons belonging to *groups*. The output of the tracking component is serialized towards the knowledge component.

The knowledge component is a java application developed using the Apache JENA framework [7], one of the most adopted framework for implementing semantic web applications. The knowledge component presents two subcomponents: one dedicated to create the instances (individuals) in the defined ontology; the other devoted to the analysis (so called analytics) for identifying several events. Using the JENA API, we manage the tracking Ontology. As soon as the output of the tracking component is available to the knowledge component, the semantic annotation process starts. For each tracked object, several inference rules are applied before the creation of instances within the on-

[3] http://schema.org
[4] http://www.w3.org/TR/html-rdfa/
[5] http://www.w3.org/TR/microdata/
[6] http://www.w3.org/Submission/spin-overview/
[7] https://jena.apache.org/

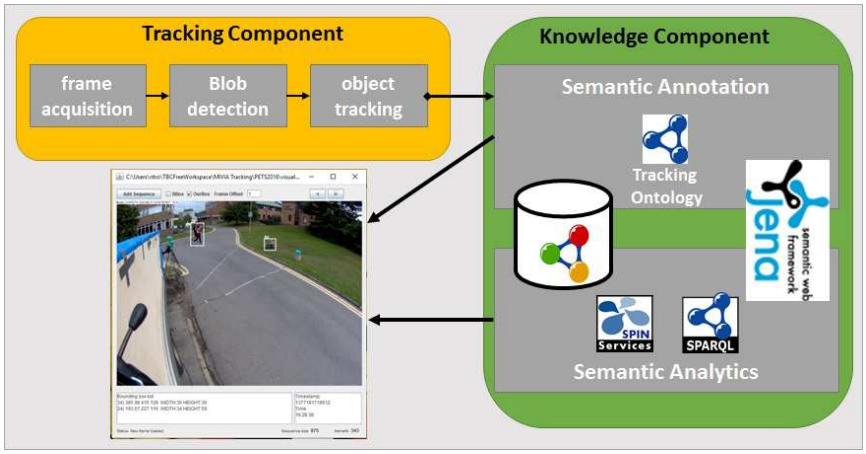

Figure 2. System components overview

tology. The inference rules using the semantic description of the scene help to identify typical mistakes of tracking algorithm like blobs generated by moving objects (e.g. trees or ribbons), object ID switches (due to splits caused by occluding objects or group joins/separations), missing objects in some frames, etc.. The semantic annotated objects are stored in the proposed tracking ontology and permanently saved in a triple store. The inference rules are expressed using SPARQL queries, where specific SPIN functions are invoked.

2.2. Our Tracking Ontology

The knowledge component is build on top of the **Tracking Ontology**. In Figure 3 the main classes of the ontology are presented. In an ontology, classes can be organized in a hierarchy and provide an abstraction mechanism for grouping resources with similar characteristic. The classes of our Tracking Ontology serve several purposes. First of all, information coming from the tracking component is semantically annotated. Every time the tracking component tracks something, the related instances are created in the ontology classes (like Frame, Blob, BoundingBox, Persons, Groups, etc.). Our ontology also codifies knowledge about the scene. Different areas are defined: those where objects can enter or exit (EntryArea and ExitArea); areas (OArea) where occluding objects are present (i.e. tree, pole, signal), classified according to their shape and, consequently, to the type of occlusion they cause. Some classes are devoted to the definition of perspective areas, that allow rules to infer information about object size according to their position in the scene (car, person, group or noise). The ontology maintains information about the different situations where tracked objects are involved (EnteringScene, LeavingAGroup, ...), including information about their movement (Walking, Running, ...) and appearance.

The Tracking Ontology also includes a set of properties relating individuals (instances of ontology classes). There exist two different kind of properties: *Object properties* and *datatype properties*. Figure 4 shows the object properties of our ontology. Object properties link individuals of a class with other individuals. For ex-

Figure 3. Classes of the Tracking Ontology

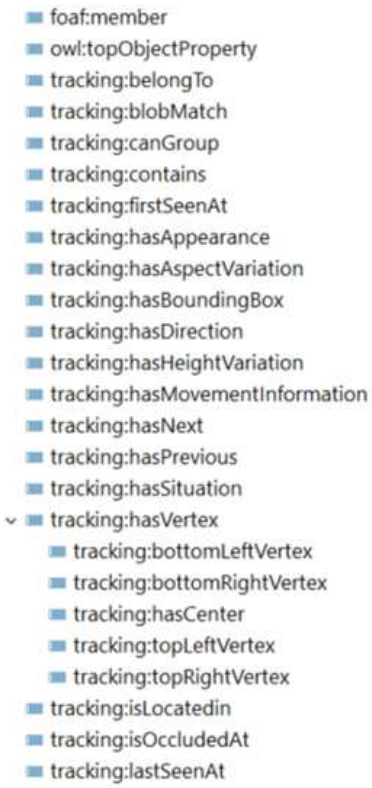

Figure 4. Object Properties of the Tracking Ontology

ample, `tracking:hasBoundingBox` is a property that links an individual of the class `SpatialThing` with individuals of the class `BoundingBox`, while `tracking:belongTo` links a specific blob to a frame where it has been identified.

Figure 5 summarizes the datatype propoerties. Datatype properties link individuals with data types. For example `tracking:hasAverageSpeed` links an individual of the `TrackedObject` class with a float representing the average speed.

The Tracking Ontology is also the basis for developing the rule-based inference system using SPIN language named **Tracking Rules**. In Tracking Rules, some functions have been defined which take as input ontology individuals and perform inference on their properties. Moreover, SPIN rules are also used inside SPARQL queries for inferencing knowledge from the annotated output of the tracking component and perform analysis for simple and complex event recognition. Listing 1 presents a SPIN rule that takes as input an individual of the class `Person` and returns the first frame where this person is falling to the ground.

In the SPARQL syntax (we recall that SPIN is based on SPARQL) variable names are preceded by the question mark (`?`) symbol. For readability reasons, in our examples we omitted the prefix `tracking` (the base URI for our ontology) before each element (class or property) of the

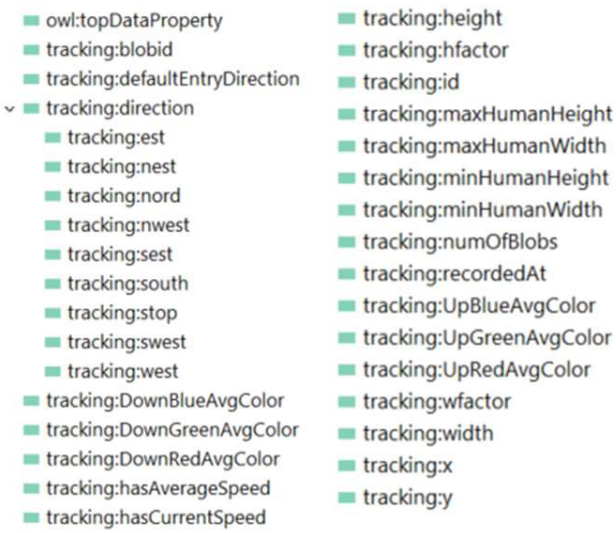

Figure 5. Datatype Properties of the Tracking Ontology

tracking ontology. The statements `SELECT`, `WHERE` and `ORDER BY` have the same meaning of the SQL language. The statement `FILTER` is used for filter the results of the `SELECT`. All the statements in the WHERE clause are intended in logical AND (all have to be satisfied). The output of the query is a sub-graph that matches the statements of the WHERE clause. The event *person falling to ground* is identified by recognizing a sequence of different situations. Considering that the tracking component identifies each object with a bounding box, we could say that a person falling to the ground changes frequently shape (from a vertical box to an horizontal one or from a rectangular form to a square one and again rectangular, etc.) for a few (1-3) seconds. After that, depending on the gravity of the fall, the person will spend some time in the same position. A SPIN rule has been defined to formalize the situation described above.

In particular, the rule starts by selecting blobs that during the annotation phase have been associated through the object property `blobMatch` to the person identified by the variable `?a_person` and belonging to a frame (`?frm`). For every matched blob during the annotation phase the current speed (`?spd`) is also calculated, associated through the datatype property `hasCurrentSpeed`. For each blob is evaluated whether a significant change in their aspect occurred or not. The second part of the rule (lines 9-14) collects the same information but for other blobs matched with the same person (`?a_person`). The last 2 lines filter the results by only taking the frames (`?frm`) where the blobs are stationary (`?spd`$<$1.0) and the blobs that in the previous 14 frames had significant aspect variations. The first frame returned by the SELECT is the frame where the event *person falling to ground* starts.

Listing 1. Rule spin:fallingGroundAtFrame

```
SELECT ?frm
WHERE {
   ?a_person tracking:blobMatch ?blb .
   ?blb a Blob .
   ?frm a Frame .
   ?frm id ?id1 .
   ?blb belongTo ?frm .
   ?blb hasCurrentSpeed ?spd .
   ?blb2 a Blob .
   ?a_person blobMatch ?blb2 .
   ?frm2 a Frame .
   ?blb2 belongTo ?frm2 .
   ?blb2 hasAspectVariation HighVariation .
   ?frm2 id ?id2 .
   FILTER ((?spd<1.0) && (?spd2>1) &&
   (?id2>(?id1-14)) && (?id2<?id1)) .
} ORDER BY (?frm)
```

This simple SPIN rule can be used in a standard SPARQL query like the one in Listing 2 to identify at which frame a person fell to the ground alone and when got-up (using two other SPIN rules `spin:getupFromFall(person, fallFrame)` and `spin:countClosePerson(person, fallFrame)`).

Listing 2. SPARQL query for identify the persons falling ground alone

```
SELECT ?pp ?frm_fall ?frm_getup
WHERE {
   ?pp a foaf:Person .
  BIND (spin:fallingGroundAtFrame(?pp) AS
      ?frm_fall) .
  FILTER
      (spin:countClosePerson(?pp,?frm_fall)=0)
  BIND (spin:getupFromFall(?pp,?frm_fall)
      AS ?frm_getup) .
}
```

3. Preliminary results

The developed hybrid system has been tested on some of the proposed PETS 2016 views for recognizing mid and high level events. In particular View RGB-2 sequence 08_3 (mid-level people falling alone) and View RGB-2 sequence 15_05 (high-level aggression to a person). Before entering into details of the experimentation, Figure 6 shows how effective is the semantic annotation. Interpreting the output of the tracking component with respect to the contextual information available in the ontology (like the position of the occluding objects, the noisy and the parking areas, together with some simple inferences about position, shape and dimension of the blob) allows to correct some of the typical mistakes made by traditional (bottom-up) tracking systems.

3.1. Mid level event recognition

The case "person falling or pushed to ground" is considered for mid level event recognition. The output of the tracking component on the 889 frames of the video sequence has been semantically annotated producing more than 95.000 triples in our tracking ontology. About 9 secs are needed to populate and store the whole ontology. Figure 7 shows the achieved results by reporting starting and ending frames for the detected event. The anticipated detection of the end of the event is due to the tracking component that estimates the position of persons inside a group by using a Kalman filter. Since the falling to ground person with ID=38 is helped by 2 other persons (ID=67 and ID=37) that join a group (at frame 390 with person 67 and at frame 405 with person 67 again), the tracking component keeps the correspondent bounding boxes at a fixed size for several frames before updating them. This implies a sudden variation in the aspect ratio that our system recognizes as the getting-up of the person. For completing the analysis of the sequence, another SPIN rule that checks whether the person fell to the ground has been helped by someone else (and in that case returns the ID of the person/s) has been defined. The execution of all the queries for recognizing the event requires about 1.2 secs, making the proposed approach especially suited for real-time elaborations.

3.2. High level event recognition

For the high-level detection case, our prototype system has been tested to detect the complex event "aggression to person". The semantically annotated tracking sequence is composed by about 92000 triples; the population and storing time is about 9 secs. Figure 8 summarizes the results of our system. The recognition of the event happens through 4 steps. Each step requires the execution of some rules of the knowledge component. The system starts with the identification of people fighting. This is done by a rule `spin:fightingAtFrame` that for each group identified by the tracking component checks the sequence of frames where a high variation of the blob aspect associated to the group is registered (using the property `hasAspectVariation`). After this step, the rule selects the groups and the frame. The results are summarized in Table 1, that reports all frames containing a possible fighting event according to our system: for each frame the identifiers of groups and involved people are also reported. In particular, four possible fighting groups are identified by the rule. Having identified groups, we can retrieve the members using the property `foaf:member` imported by FOAF (Friend of a Friend) schema used for describing persons, groups and their attributes.

The second step checks for every person member of the group their behavior in the next frames, verifying whether they run away or stay stationary. According to the rules

Group ID	Frame ID	Person ID
9	831	1,3
26	1003	1,2,3,4
30	1036	1,3,4
31	1056	1,2,3,4

Table 1. results of the fighting rule

Figure 6. Left image: output of the tracking algorithm with a lot of blobs produced by the camera moving; right image: output of the semantic annotation component applying filtering rules

devoted to check this sub-event, the system infers that (for the Group-9 starting from frame 831) person-1 and person-3 do not run. For Group-26 we have that the persons 1,3 and 4 run for most of the frames, while person 2 spends all the time stationary (except for a couple of frames, where movement is actually due to an ID switch generated by the tracking component). Information obtained by applying the rules in sequence allows us to infer that the event aggression starts at frame 1003. It is worth to mention that person-2 is member of both Group-26 and Group-31. This is due to a mistake of the tracking component that misses person 2 on the grass for some frames and assigns the ID-2 to one of the member of Group-31. The final step is devoted to the identification of the frame where the aggression ends. With the proposed knowledge based system we can identify the end of the aggression by defining different rules. We decided to set the last frame of the aggression as the frame where the aggressors (running persons) leave the scene. The identification of this frame is straightforward since in the ontology we have two properties `firstSeenAt` and `lastSeenAt` that record the first and the last frame when a tracked object is seen in the scene.

It is worth mentioning that the analysis of the scene through our hybrid approach does not cause any false positives. Indeed, we tried to recognize both the events (person fall to the ground alone and aggression to person) on both the considered sequences and only the two events of interest have been correctly identified.

4. Conclusions

In this paper an hybrid approach for performing high level events recognition using semantic web technologies has been presented. The achieved results are encouraging both in terms of accuracy and performance. Semantic annotation of traditional tracking algorithm output (objects, frames, blobs and bounding boxes) allows on one hand to produce reliable results even in presence of classical tracking errors (IDs switch, noises blobs, etc.) and on the other hand the creation of powerful video analytics solutions even on large volume of data as needed for big data analytics.

Due to the strong dependence from the quality of the output produced by the tracking algorithm, next steps will be devoted to make more robust the ability of the knowledge component to correct the tracking mistakes trying also to infer information considering the output of the detection phase.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. 2

[2] J. Ferryman and A.-L. Ellis. Performance evaluation of crowd image analysis using the {PETS2009} dataset. *Pattern Recognition Letters*, 44(0):3 – 15, 2014. 1

[3] P. Foggia, G. Percannella, A. Saggese, and M. Vento. Real-time tracking of single people and groups simultaneously by contextual graph-based reasoning dealing complex occlusions. In *Performance Evaluation of Tracking and Surveillance (PETS), 2013 IEEE International Workshop on*, pages 29–36, 2013. 3

[4] A. Franois, R. Nevatia, J. Hobbs, and R. Bolles. Verl: An ontology framework for representing and annotating video events. *IEEE Multimedia*, 12(4):76–86, 2005. 2

[5] J. Gomez-Romero, M. A. Patricio, J. Garca, and J. M. Molina. Ontology-based context representation and reasoning for object tracking and scene interpretation in video. *Expert Systems with Applications*, 38(6):7494 – 7510, 2011. 2, 3

[6] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5):907–928, 1995. 2

[7] R. D. Lascio, P. Foggia, G. Percannella, A. Saggese, and M. Vento. A real time algorithm for people tracking using contextual reasoning. *Computer Vision and Image Understanding*, 117(8):892–908, 2013. 1

[8] D. L. McGuinness, F. Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004. 3

[9] G. Meditskos, S. Dasiopoulou, V. Efstathiou, and I. Kompatsiaris. Sp-act: A hybrid framework for complex activity recognition combining owl and sparql rules. pages 25–30, 2013. 2

[10] L. Patino and J. Ferryman. Multiresolution semantic activity characterisation and abnormality discovery in videos. *Applied Soft Computing*, 25:485 – 495, 2014. 1

[11] S. K. Reed. *Cognition: Theory and applications*. 9th edition, 2011. 2

[12] D. Riboni and C. Bettini. Owl 2 modeling and reasoning with complex human activities. *Pervasive and Mobile Computing*, 7(3):379–395, 2011. cited By 34. 2

Figure 7. Person fall to the ground alone event on View RGB-2 sequence 08_3 dataset: the result of our system compared to the ground truth. The picture on the left shows the rendering in our analysis tool of the output of the tracking component at the frame identified by the knowledge component as the first where a person is (starting) falling to the ground alone. The picture on the right reports the output of the tracking component at the end of the event. The numbers marked as ground truth are the frame numbers where the events start and finish respectively.

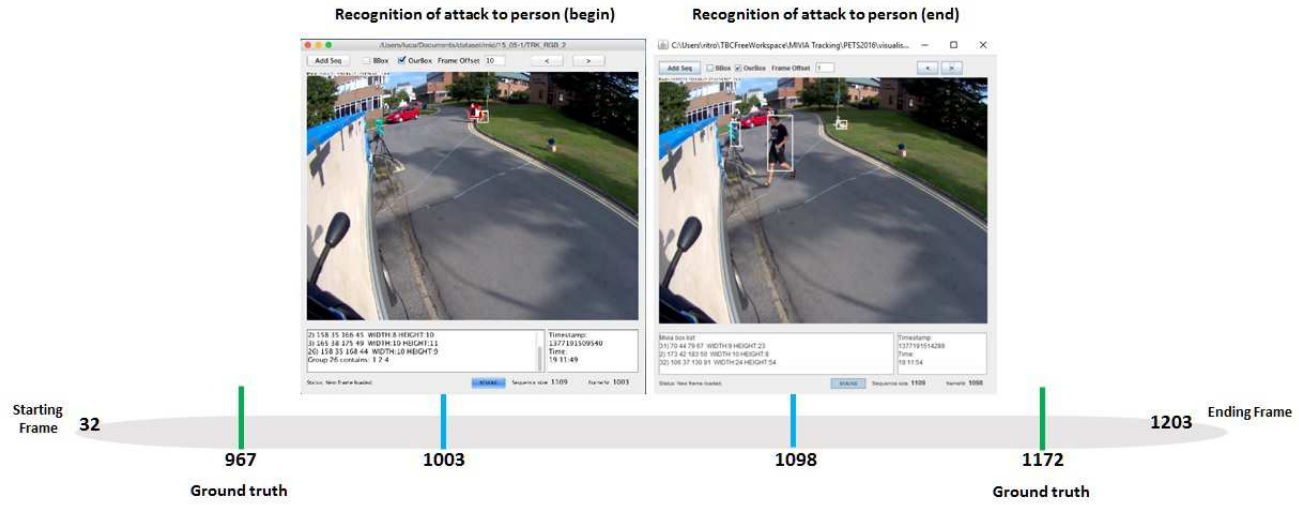

Figure 8. Attack to person event on View RGB-2 sequence 15_05 dataset: Results of our system compared to the ground truth. The first picture (left) shows the rendering in our analysis tool of the output of the tracking component at the frame identified by the knowledge component as the first where the aggression to a person starts. The second picture (right) shows the output of the tracking component at the end of the event. The numbers marked as ground truth are the frame number where the events start and finish respectively.

[13] J. SanMiguel and J. Martnez. A semantic-guided and self-configurable framework for video analysis. *Machine Vision and Applications*, 24(3):493–512, 2013. 2

[14] G. Sanrom, L. Patino, G. Burghouts, K. Schutte, and J. Ferryman. A unified approach to the recognition of complex actions from sequences of zone-crossings. *Image and Vision Computing*, 32(5):363 – 378, 2014. 1

[15] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006. 1