This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS

Semantically Enhanced UAVs to Increase the Aerial Scene Understanding

Danilo Cavaliere, Vincenzo Loia, *Senior Member, IEEE*, Alessia Saggese, *Member, IEEE*, Sabrina Senatore, *Member, IEEE*, and Mario Vento, *Member, IEEE*

***Abstract*—Visual tracking supported by unmanned aerial vehicles (UAVs) has generated a lot of interest in recent years, especially in application domains such as surveillance, search for missing persons and traffic monitoring. The major challenges in visual tracking with small UAVs arise in the form of target representation, target appearance change, target detection and localization in real time computation. Reliable target detection depends on factors such as occlusions, image noise, illumination and pose changes, or image blur that may compromise the object labeling. To mitigate these issues, this paper proposes a hybrid solution: along with the tracked objects, scenes are completely depicted by adding contextual information, i.e., data describing places, natural features, or in general points of interest. Each scenario indeed is semantically described by ontological statements that define the context and then, by inference, support the object tracking task in the object identification and labeling. The synergy between the tracking methods and semantic modeling can bridge the object labeling gap, enhancing the scene understanding and awareness when alarming situations are discovered. Experimental results are promising and confirm the applicability of the proposed framework in supporting drones in object identification and critical situation detection tasks.**

***Index Terms*—Mobile camera, semantic Web, situation awareness, unmanned aerial vehicles (UAVs), video tracking.**

I. Introduction

IN THE recent years, aerial surveillance is becoming crucial in many safety-critical application domains, such as fire detection, traffic congestion or accidents, etc. Unmanned aerial vehicles (UAVs) represent a clear, low cost reply to ground-plane surveillance systems, in order to recognize alerting situations. Although UAVs should guarantee rapid time of response, especially when considering a victim's mortality and morbidity after a severe injury accident, at the same time, they should also potentially fly in uncomfortable weather conditions, that could be too dangerous for a manned aircraft. The main issue with the UAVs is the difficulty in acquiring a high-level description of the scenes appearing in the video sequence, only from the object detection, identification and tracking algorithms. Enabling a UAV to acquire a complete description of the scenario, during the flights and then, to recognize critical scenes from a video sequence is indeed, a very useful and desirable capability. To this purpose, a robust tracking to handle complex tasks such as object identification [1] and event detection [2] is required. Many studies focus indeed on alleviating common problems related to UAV video tracking such as camera resolution, camera shaking, illumination change and appearance change of the background [3], [4]. but also to achieve optimal trajectory tracking [5] and alleviate vehicle routing problems [6]. Although many tracking algorithms can deal with occlusion, split objects, shadows, and reflection, object tracking suffers in object labeling [7]; moreover, camera movements add further problems to the object tracking algorithm. Most of the algorithms presented in literature work on object tracking with a fixed camera, and, on moving camera, the traditional background subtraction algorithms are not applicable. For this reason, most approaches concentrate on a single object class recognition task, for instance, pedestrians [8] or crowds [9]; vehicles [10], or their relations [11] and the main applications in this area converge on a single type of scenario [12], In the light of these observations, our basic idea is to improve the object tracking task in a video sequence, augmenting the tracking data with the contextual data, i.e., complementary data related to the surrounding background objects of the scene, in order to acquire a more complete scene information to alleviate tracking issues. Data from tracking algorithms and additional background information are collected and coded into ontological statements: the role of the semantic Web technologies in the modeling of tracked objects and their relations with other objects in the environment is critical for object classification and labeling, especially when a moving camera is involved (videos came from a flying UAV).

The rationale behind the semantic enrichment of the scene description is indeed to exploit the contextual information from surrounding background objects such as places, buildings, rivers, and roads, in general, points of interest (POIs) to better identify and then label the tracked objects. Just to give an example: a tracked object on a road (where other similar tracks appear), more likely is a car; if a similar track appears on a river, it will be a boat. An object trajectory (hereinafter

Manuscript received August 1, 2017; accepted September 17, 2017. This paper was recommended by Associate Editor Z. Li. *(Corresponding author: Vincenzo Loia.)*

D. Cavaliere, A. Saggese, S. Senatore, and M. Vento are with the Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata, University of Salerno, 84084 Fisciano, Italy (e-mail: loia@unisa.it).

V. Loia is with the Dipartimento di Scienze Aziendali—Management and Innovation Systems, University of Salerno, 84084 Fisciano, Italy.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This includes videos that show articulated situations set on the road with some critical events. This material is 34.1 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2757462

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS

a *track*) whose predominant color is red could be a fire: if it appears in a wood, then probably it represents a dangerous situation; but if it appears on a beach, it could be just a bonfire, which is likely not dangerous.

Semantically coded scenes feed a knowledge base, which becomes a source to query and to infer comprehensive, high-level information about the objects enclosed in the scene and in the video. The ontology-based modeling of scene from video sequences can enhance the video tracking methods, supporting the object labeling and providing a high-level interpretation of the scenes in the video sequence: objects are discovered and automatically labeled with the actual name; at the same time, event and object interactions in the scene can be monitored so that a critical situation can be detected when an alarming event (pedestrian on the road, car crash, fire, etc.) is revealed. This paper presents a novel approach which exploits the synergy between the tracking methods and semantic technologies to bridge the object labeling gap, enhance situation awareness, as well as detect and classify alerting situations. The UAV, provided with a camera mounted on board, can recognize moving and background objects, that populate the scene, and relations/interactions between them. The semantic technologies provide the way to collect all these data and produce a comprehensive description of the scene, that will be used to infer additional information. The UAV becomes "aware" of the situations occurring in the evolving scenario and, from the contextual (background) data, can also individuate and interpret alerting situations. In a nutshell, the proposed study presents the following.

1) An approach that merges data from video tracking output (by a mobile camera) with high-level environmental data, in order to provide a straightforward scene description and understanding.
2) A semantically enhanced knowledge base which collects data describing dynamic scenarios involving various kinds of objects and environments.
3) A reasoning process to produce additional contextual knowledge (feeding the knowledge base), useful to provide object classification and event detection.

This paper is organized as follows. Section II provides a brief overview about the main works on video tracking, and the role of semantic technologies in supporting object tracking and labeling approaches. Section III sketches the overall framework model; then Section IV introduces the semantic model on which the framework lays; specifically, a closer look at the semantic modules and their interactions is given in Section V. Experiments in Section VI show the effectiveness of this approach in the object classification and labeling as well as the scene understanding. The conclusion and future work close this paper.

II. Related Works

Although high-level knowledge-based scene understanding from moving camera is a hot topic for improving decision-making processes and supporting video-tracking activities like moving object detection and tracking [1], [2], [14], there are a few related works in literature studying the problem. This depends on the fact that the moving camera causes a lack of reference points in the scene which affects both object detection and tracking activities, and high-level scene modeling. Then, most of studies focus only on low-level data coming from video or at most adding only few environmental variables to the problem. Moreover, *a priori* knowledge based on static context is not suitable with a dynamic environment like a scene taken from a moving drone with an on-board camera, which can record many different environments with many different moving object kinds moving in it.

The moving camera adds new problems to object detection and tracking, especially because there is no fixed background, which makes the distinction between self moving objects and environmental elements more difficult [15]–[18]. Therefore, consolidated fixed camera techniques like background subtraction [19] cannot work because environmental element pixel data change with the moving camera. To constrain this problem, studies on the moving camera video-tracking make assumptions on the environment and camera; for instance, they assume *a priori* that the environment is finite and well known [12], [20], [21]; camera movements are constant or constrained; tracking is carried out on only one object [8], [10]. Some studies also achieve object recognition by object classification in predefined classes, even though many issues as low resolution [22], motion blur [23], and prohibitive camera shots [24], [25] need to be addressed.

The goal of our approach is not a new object tracking algorithm but a knowledge-based model to understand scene dynamics at a high level, and to better improve object classification, as well as to support complex decision tasks. For scene understanding, many works propose pattern recognition methods to recognize scene elements or regions [26]. Classification results are quite limited and do not provide a deeper and high-level understanding of the scene, which is required when dealing with evolving scenarios. Generally, most of these methods work exclusively on low-level pixel-based data, such as color, shape, and position. For a better contextual awareness of the scene, it is necessary to retrieve and model the environmental data. Environmental data can be used to build an initial knowledge on environment, useful to contextualize features and the moving objects. Studies [27] have proposed approaches that build spatio-temporal contexts exclusively on the tracking data.

In order to get a deeper knowledge of the environment, this paper presents a framework adopting semantic techniques to model a dynamic environment starting from some basic features. Semantic technologies allow building a high-level description of the environment and its elements, based on heterogeneous information gathered from different sources. They provide a machine-oriented representation of the scenario and the situations evolving in the scenario. It is the main role, along with the inference process that, applied to the built model, can enhance the knowledge about the evolving scenario. The enhanced knowledge is useful to understand complex situations in the current scenario, even though it is far from the comprehensive prediction of possible dangerous situations, especially when unpredictable behaviors happen, nor it is able to quantify the chance of an event occurring [28]. A synergistic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS

approach that exploits consolidated methodologies (for example, deep learning-based methods) could alleviate the issues related to the foreseeing of future unexpected/unpredictable events.

In the light of these observations, our idea is to exploit the semantic technologies to provide a semantic enhancement in the object labeling and scene understanding, in order to suggest critical situations (i.e., situations where for example, the spatial relations among objects are out of allowed range) and eventually, to wisely support a decision. Semantic technologies are also used in video-tracking but with fixed camera. They are mainly used for data fusion of low-level data coming from different sensors [29], and for data fusion between low-level data and contextual variables [30]. The main goal of methods proposed in literature is to alleviate tracking problems like occlusion, grouping, shadowing, etc. [30]–[33]. They are also used to support object detection proposing semantic segmentation techniques to classify pixel regions in predefined classes [34]. Semantic segmentation often involves deep learning techniques to classify pixels in predetermined categories. These methods report good performances on detecting environmental features [35], even though they require many preacquired training samples [36]. In [30], a framework producing high-level knowledge on an environment is presented. The application recognizes a door, a person by dimensions and the action of the person entering in the scene by the door. This approach needs to acquire the scene *a priori* (e.g., door presence) with a fixed camera filming a static and well-known environment. Our approach, instead, is aimed at building an adaptable framework for various possible scenarios: it models at the semantic level, first, basic and general concepts, suitable for every kind of scenario, and then, employs a map-based tool to retrieve more specific environmental data, to enrich the knowledge about the scenario. To the best of our knowledge, there are no studies on the moving camera video tracking employing semantic Web technologies to improve the accuracy in the object identification. Moreover, this hybrid approach can overcome the missing data problem in tracking algorithms, and, thanks to the expressive power of the semantics, produces a high-level description of events and objects in the scene.

III. Framework Overview

Fig. 1 shows the high-level scheme of the framework, with all the components and their main interactions. The core of the framework is represented by the semantic modules that, in the figure, are enclosed in a black border square. This framework extends a preliminary work, presented in [37].

The main input is a video recorded by a flying drone with an on-board installed camera (top left of Fig. 1). The recorded video is taken as input by the *tracking module* which extracts the trajectories of the objects moving in the scene, frame by frame. For each frame, tracked object dimensions and speeds are also calculated. The other input of the framework is environmental data (top right in Fig. 1): it is composed of specific places called POIs, which are fixed geo-referred points or areas retrieved with Google Maps service, lying in the area where

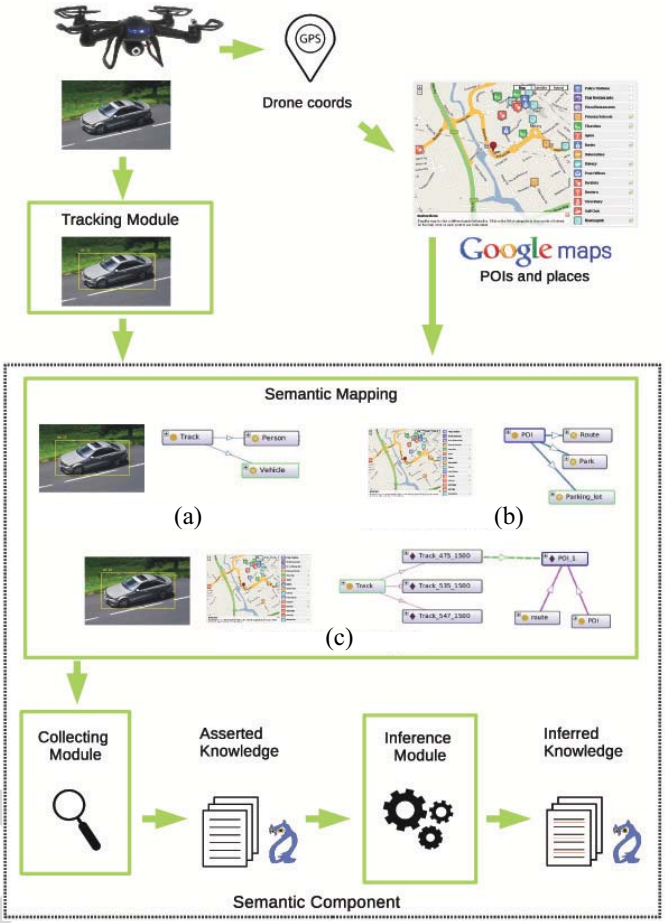

Fig. 1. Logical overview of the framework.

the drone flies. The video sequence and the object trajectories, as well as the POIs retrieved with Google Maps are passed to the *semantic mapping* module. This module in turn, translates tracking and contextual data in semantic statements according to TrackPOI ontology, an ad-hoc designed ontology to model the on-the-road scenarios. The *semantic mapping* module is composed of three submodules. The first one is *track semantic mapper* which maps moving objects and frame data in assertions about their identity, real dimensions, speed and position. The *POI semantic mapper* aims at defining assertions on the POIs data retrieved with Google Maps query. The third module is *relation semantic mapper*: from the knowledge base produced by the track and POI semantic mappers, it extracts positional relations between tracked objects, and tracked objects and POIs in the scene, and then generates the corresponding assertions which feed the knowledge base.

The collected knowledge on tracks, POIs, and their relations is passed to a *collecting module*, which collects and selects most relevant assertions from the semantic mappers to better model and refine the contexts in the evolving scene. Finally, this knowledge is passed to the *inference module*, which deducts new assertions on tracks, POIs and relations, to feed a comprehensive knowledge base, for a deep high-level scene understanding.

Before detailing the framework component, a brief description of the ontology and its role in the semantic description of main components is given in the next section.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS

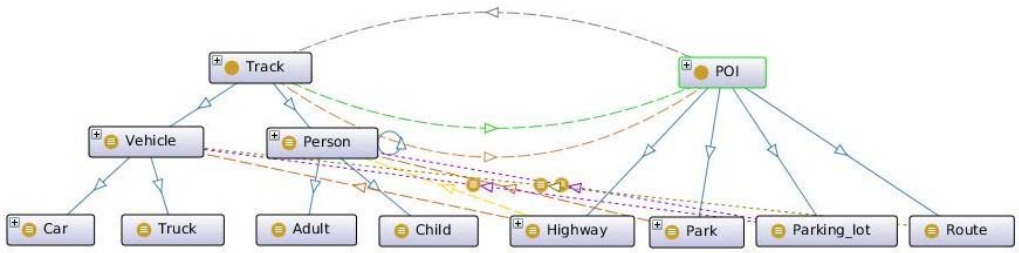

Fig. 2. TrackPOI ontology schema.

IV. Ontology Design

The ontology design provides an explicit, shareable way to describe a domain of interest. In this paper, it describes geo-contextualized scenarios coming from video sequences, recognizing not just the single entities appearing but interpreting the whole evolving scene.

Fig. 2 shows TrackPOI ontology schema. The ontology presents two main classes *Track* and *POI*, which, respectively, model track and POI entities. Track class presents two subclasses: 1) *Person* and 2) *Vehicle*. These subclasses are the possible types of moving objects present in the scene and each one of them has other subclasses, for *Vehicle*: car, motorcycle, and truck; and for *Person*: adult and child.

POIs are structured data about urban and natural environments (e.g., roads, bridges, squares, buildings, stores, institutions, mountains, oceans, rivers, etc.) and localized by their own GPS position. In Fig. 2, only the subclasses *Highway*, *Park*, *Parking_lot*, and *Route* are shown. Besides the class-subclass relation, transversal relations are also shown; a high-level generic relation *hasRelationWith* between *Track* and *POI* is given: it states that some relationship exists between some track and some POI. Specialized relations can be defined increasing the subclass depth, where well-designed relations link subclasses; for instance a specialized relation *isinTheAreaOf* exists between classes *Person* and *Park*. Further details about the designed relations are given in the following sections.

V. Deep Outlook on Semantic Modules

Although the section presents all the components involved in the framework modeling, the main focus is given to the modules embedding semantic processes.

A. Tracking Module

The tracking module extracts the trajectories of those objects moving in the scene, and provides to the *semantic mapping* module the position (and the dimensions), frame by frame, of each object moving in the scene. The adopted tracking algorithm was proposed in [38]. It is based on three steps.

1) *Camera compensation*, which aims at estimating the movement of the camera so as to extract a kind of background of the scene representing the scene with no objects inside.
2) *Detection*, which extracts the position of the objects moving in the scene at the current instant of time.
3) *Tracking*, which aims at associating the objects detected at the current frame with the ones tracked up to the previous frames, so as to draw, for each object, their trajectories frame by frame.

In cases of a low probability of this association, a feedforward approach is exploited: the parameters adopted during the detection step are indeed adaptively and automatically adjusted, so as to increase the probability of the association. The tracking algorithm is able to effectively deal with both merging (two or more objects identified as a single one during the detection step) and splitting (a single object partitioned into two or more parts during the detection step). Let us remark that the tracking algorithm has been just used to get tracked objects in a format suitable to be processed by our semantic components; its performance analysis is out of the topic of this paper and is presented in [38].

B. POI Semantic Mapper

The POI semantic mapper processes POIs and places appearing in video scenes. POIs are fixed objects in the scene by definition, and for this reason they can be considered as reference points, which are very useful to understand movements of objects present in a moving camera recorded video. Furthermore, POIs play another important role in modeling knowledge about the video scenes. In this paper indeed, POI data are also used to define a context by restricting domain about the scenario. For example, if the system retrieves POI data about a public park which generally not allows vehicle transit, the main moving objects walking and standing in the park area will be people and pets.

Google API provides Google Maps Geocoding API[1] and Google Maps Places API[2] to localize, geo-refer and retrieve specific data about POIs. Therefore, POI semantic mapper adopts Geocoding API to make reverse geocoding by a simple query, which takes a pair of coordinates and returns a Json/XML file with a human-readable address and related data about the area which the pair of coordinates corresponds to. The main retrieved data are POI identity or type, administrative area level, postal code, street address, and GPS area and position. For each retrieved POI, the POI semantic mapper generates a new POI individual, i.e., an instance of class *POI*

[1] https://developers.google.com/maps/documentation/geocoding/intro
[2] https://developers.google.com/places/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS

(described by *TrackPOI* ontology, Fig. 2), and codes its own retrieved data in RDF triples. Precisely, the POI position and area are retrieved from *geoRSS* ontology,[3] a well-known geographical ontology that provides geospatial properties of POIs, and then, they are integrated into our ontology.

Since Geocoding API returns data about macro places like university, park, zoo, etc., Google Maps Places API has been queried to get information about small and more simple POIs, which are additional reference points which a track can also interact with. Google Maps Places API indeed returns a list of 97 different places (e.g., bank, bar, and park) and their related information in a similar way to Geocoding API.

Let us notice that a place is not associated with an area (like macro places) but just a position identified with GPS coordinates. Queries submitted to both Google APIs are based on the drone coordinates (associated with every frame of the recorded video), whereas the covering radius to retrieve places is based on the distance covered by the drone.

In a nutshell, the POI semantic mapper retrieves data about the POI location, identity and its related data which contribute to depict the context of a scenario. Then, each POI appearing in a frame is coded as a POI individual, i.e., a class instance of the *TrackPOI* ontology.

C. Track Semantic Mapper

The *Track* semantic mapper is in charge of converting the tracking output, provided by the tracking module, in ontological statements. A tracked object or simply a track is a dynamic object present in the video, that moves in the scene. A track can rapidly change from a moving to fixed state and viceversa in a few of frames. In most cases, tracks move autonomously in the environment, and they often are living beings like humans or animals, but they can also be not living things carried, pushed or driven by living beings like vehicles, balls, etc. The study focuses on various outdoor environments mainly populated by humans, animals and vehicles. The tracking output includes data about the tracks and frames which they are in. The *Track* semantic mapper reads the tracking output and extracts data about each track. For each track identified, the mapper generates a *Track* individual according to the proposed TrackPOI ontology (Fig. 2), where *Track* is the ontology class modeling the tracked objects. The *Track* individual describes a specific track in the frame, by means of the following properties: bounding box ID, track name, track real dimensions (width, height) in meters, track speed, track position (GPS coordinates), and frame ID which the track is related to. Another important added property is the *TrackPOI*:*hasRelationWith*, which relates the track individual to all the other tracks or POIs present in the same frame. All the generated *Track* individuals along with their own properties are added to the knowledge base.

D. Relation Semantic Mapper

The relation semantic mapper acquires the asserted knowledge on both tracks and POIs, generated by the other two semantic modules, POI semantic mapper and *Track* semantic mapper, respectively. Its goal is to recognize relations between the distinct entities appearing in the frames by the analysis of the possible interactions between them in the evolving scene. The investigated relations are between a track and a POI, and between two tracks representing different entities. Relations are coded as asserted knowledge and are crucial to delineate the context of the scenario. The study focuses particularly on the positional relations, suitable to analyze how a track is positioned with respect to fixed POIs in the scene, and how a track is interacting with other tracks in the scene. The analyzed relations cover the main grammar prepositions of place: *front of, behind, near, between, in,* and *on*. Each preposition describes a geometric relation between GPS coordinates of tracks and POIs.

As a first task, relation semantic mapper creates a relation instance *TrackPOI*:*hasRelationWith* between each track and POI, present in the same frame, then it adds them to the asserted knowledge. The mapper processes track-POI relations frame by frame, in order to specialize these relations with respect to the contextual information. For each frame indeed, it analyzes every *TrackPOI*:*hasRelationWith* statement among track-track and track-POI filtering out irrelevant relations (i.e., on tracks without a specified position).

Geometric calculations are applied on each relation per frame, taking into account: GPS coordinates of the tracks and POIs, positional speed of a track in the current and previous frame, track and POI real dimensions and track directions. The geometric data allows relation semantic mapper to discern positional relations between entities in the scene, for example, a track in the area of a POI, a track/POI in proximity of another track/POI, etc. More specifically, relation semantic mapper can produce six different specializations of the predicate *TrackPOI*:*hasRelationWith*, that are modeled in the TrackPOI ontology. They are detailed as follows.

1) *TrackPOI:isInTheAreaOf:* This predicate is used to relate two entities, x and y. The statement x *TrackPOI:isInTheAreaOf* y [see Fig. 3(a)] is produced if the GPS coordinates of x (a track or a POI) lie within the area of y (a POI). Generally, the area of y is retrieved by the Google Maps query on POIs, that provides two pairs of coordinates which, respectively, represent the north-east and south-west zone bounds. When the retrieved POIs lack of these area bounds, the relation semantic mapper defines a covering radius, according to place type and video features, to delineate an area for the POI.
2) *TrackPOI:isNear:* This predicate relates two entities x and y which are close to each other, and, similarly to predicate *TrackPOI:isInTheAreaOf*, the mapper enables us to specify a covering radius value to define the concept of closeness [see Fig. 3(b)].
3) *TrackPOI:hasDirection:* This is a predicate that can be specialized, according to spatial relations between two entities. The mapper calculates the track trajectory (considering positions in successive frames) which is translated in cardinal points; an assertion x *TrackPOI:hasDirection* p is produced for every frame

[3] http://www.georss.org/rdf_rss1.html

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS

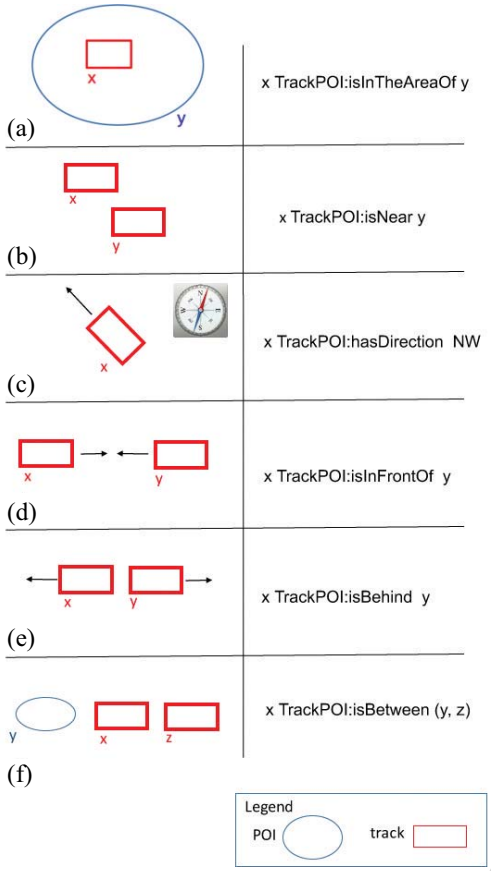

Fig. 3. Specializations of the predicate TrackPOI:hasRelationWith, according to the geometric relations among entinties in the scene.

the track is in, where x is the track and p the cardinal point representing its direction [see Fig. 3(c)].

4) *TrackPOI:isInFrontOf:* Thanks to track direction per frame, the relation semantic mapper evaluates if two entities x, and y, are coming in front of one to the other. In detail, the assertion *x TrackPOI:isInFrontOf y* holds if the direction of x in a frame is opposite to the direction or position of y, where y is a POI and the distance between x and y in successive frames decreases [see Fig. 3(d)].
5) *TrackPOI:isBehind:* In a similar but reverse way, the mapper uses the predicate *TrackPOI:isBehind* to state x and y leave each other behind and proceed to opposite direction ways [see Fig. 3(e)].
6) *TrackPOI:isInBetween:* If the track x is between two entities y and z, relation semantic mapper can state that *x TrackPOI:isInBetween (y, z)*. This assertion holds if the position of x falls on the conjunction of y and z direction vectors, and if also the assertions *x TrackPOI:isNear y*, *x TrackPOI:isNear z* hold [see Fig. 3(f)].

E. Collecting Module

The collecting module completes the assertion process, taking as input the asserted statements produced by the semantic mappers. It syntheses the semantic knowledge removing redundant information, then checks if the statements are consistent, according to TrackPOI ontology and drone data; finally, statements are merged producing the definitive assertional knowledge base (ABOX). The asserted statements are conform to class and property definitions and restrictions of TrackPOI ontology (Fig. 2). The collecting module checks if these restrictions are satisfied to guarantee a consistent schema, especially on the relations between track and track or track and POI. For example if a track x has a *TrackPOI:inFrontOf* relation with another track y, (*x TrackPOI:inFrontOf y*), y must have the same relation with x (*y TrackPOI:inFrontOf x*) because the *TrackPOI:inFrontOf* property holds on two objects going toward each other. *TrackPOI:inFrontOf* is indeed a symmetric property.

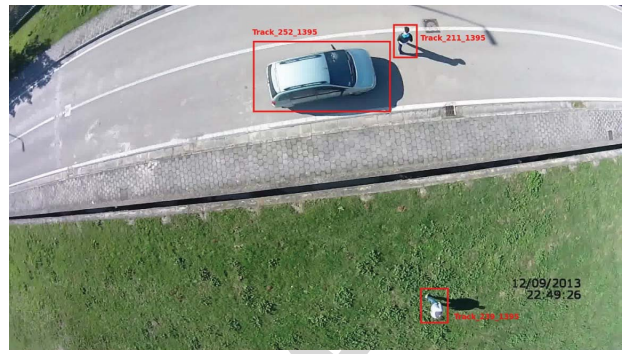

Fig. 4. Frame 1395 from video 1.

The statements produced are merged to form the definitive asserted knowledge base on all the scenes of the video.

In order to give an example about the way collecting module generates the comprehensive assertional knowledge, some statements from a video sample are discussed. Fig. 4 shows a video frame (whose identifier is 1395), showing two persons: one walking on the grass, and the other one crossing the road, in the proximity of a moving car. The frame shows the corresponding two bounding boxes, identified by the tracking module. Algorithm 3 describes all the tracks appearing in the frame by ontological statements (black rows 1–32 and 38–50). For the sake of simplicity, the statements in the form of triples <*subject-predicate-object*> are expressed in Turtle[4] semantic language. Rows 6–18 show statements on the POI entity named *POI_1*: it is an individual of the *TrackPOI:POI* class (row 8); it is a route (row 9). It is also described by assertions on the area and the geo-position (rows 16 and 17), and other Google Maps data like street address, country, administrative area level, and postal code (rows 10–15). Tracks are described in black rows 20–32 and 38–50, precisely track_211_1395 and track_252_1395 whereas the statements about track_239_1395 are not shown because very similar to axioms about track_211_1395. The track objects are individuals of the class type *Track* (rows 21–22 and 39–40) with object dimensions and movements data like width, height, and speed (rows 28–30 and 46–48) and GPS track position (rows 31 and 49). The remaining assertions on track entities are about movements and relations. Specifically, statements in rows 23–27 and 41–45 describe the geometrical or positional relations that the track has with other tracks and POIs in the

[4]https://www.w3.org/TR/turtle/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS


```
1 Track
2  and ((hasRelationWith some (POI
3  and ((ObjectsAllowed some Person) or (ObjectsAllowed only Person)))
       )
4  and (hasHeight only xsd:decimal[>= 100 , <= 220])
5  and (hasSpeed only xsd:decimal[>= 0 , <= 37])
6  and (hasWidth only xsd:decimal[>= 10 , <= 90]))
```

Listing 1. Equivalent class restriction for Person class.

same frame. Direction statements express the track direction from the previous frame with compass points (rows 32 and 50).

The knowledge base about each video frame is passed to the inference module, which, by the analysis of the acquired statements, can infer new statements about the scene.

F. Inference Module

The inference module implements the reasoning component of the framework. As stated, it produces new axioms about the scene objects and the context built on their relations, with the aim of enriching the knowledge base and enhancing the scene understanding. The new knowledge produced by this module is aimed mainly at providing object classification and labeling, as well as recognizing critical events in scene sequences. The inference engine is built on OWL class equivalent, subclass and disjoint restrictions as well as on rules implemented with Semantic Web Rule Language (SWRL).[5]

Class restrictions are useful to provide precise modeling of classes to guarantee a straightforward reasoning process in inferring new assertions and provide their accurate classification. Track individual classification has been designed on *Track* class restrictions which involve the main bounding box features and contextual data. Track subclasses (e.g., person, vehicle, etc.) are defined as equivalent class restrictions, that express a high-level definition of the class type (see Fig. 2). Algorithm 1 shows an example of *Person* class modeling as a class restriction: an individual of the class *Person* requires real dimensions (width, height) falling in a specific range which reflects the precise dimensions of a person seen from a top view (rows 4 and 6), and a speed which is acceptable for a moving human being (row 5). Relations with the context have to be also specified: the *Person* class definition requires at least a relation with a POI which admits persons in its area (e.g., the presence of the person in a park). The admissibility about the presence of a *Track* individual in a POI area is expressed by the *ObjectsAllowed* property (rows 2, 3). In order to define the right relations between entities, allowable *Track* individuals for a certain POI have to be specified. Algorithm 2 for instance, outlines that the only allowable *Track* types for a *Park* class are individuals of the *Person* class: only person tracks can appear in a park area. If an object cannot be related to some POI (e.g., a person is not in a Park), the object can be recognized by its dimensions and speed, but it is marked as not recognized by context with a special property.

Restriction-based reasoning supports the object labeling, but the expressive power is not suitable for situation understanding. Rules have been designed to recognize situations and

[5]https://www.w3.org/Submission/SWRL/

```
1 POI and (ObjectsAllowed only Person)
```

Listing 2. Equivalent class restriction for park class.

consequently alerting events occurring when restrictions on the scenes involving track objects and POIs do not hold. Each rule has been designed to verify that, in a specific situation, no unexpected event is revealed. When it is triggered, it identifies the critical event occurred on the involved objects and forces the system to provide an alert. For example, let us consider the following rule:

$$Person(?x) \wedge Route(?y) \wedge Vehicle(?z) \wedge isInTheAreaOf(?x, ?y) \wedge isInTheAreaOf(?z, ?y) \wedge isNear(?x, ?z) \rightarrow isInDangerOn(?x, ?y).$$

The SWRL rule describes an alerting situation that happens when a person is on a road. Given a *Route* individual y, and two tracks, respectively, representing a *Person* individual x and a *Vehicle* individual z, if x falls in the area of y, ($isInTheAreaOf(?x, ?y)$), the person x is in danger on the route y, especially because the vehicle z is coming ($isInTheAreaOf(?z, ?y) \wedge isNear(?x, ?z)$). The inferred property *isInDangerOn* represents an imminent alerting situation, a kind of prevision of a future dangerous situation. The reasoning process, mainly based on class restrictions and rules, works on one frame at a time. The inference module cycles on frames: it retrieves the statements related to it with a SPARQL query, and produces a subset of statements associated with each frame. Then, the inference module processes this statement subset and infers new statements, which are added to the knowledge base. Thus, when the inference module processes the statements for the frame in Fig. 4, new assertions are generated on the tracks Track_211_1395 and Track_252_1395: Algorithm 3 shows the final augmented knowledge, with an emphasis on the new assertions, red colored in Algorithm 3.

These assertions state that Track_211_1395 is actually a person (row 35): so far the previous asserted statements (rows 21, 22) state that it was just a track and thanks to inference module, Track_211_1395 is labeled as a person. Track_252_1395 is a vehicle (row 52) and more specifically a car (row 53). Statements on the situation are also deducted: Track_211_1395 is in an alerting situation, since it is on the route POI_1 (row 36).

VI. Experimental Results

An experimentation has been conducted on some videos, in order to evaluate our system performance in terms of object classification and capabilities in producing a complete scene understanding. The framework has been tested on full recorded videos, selecting consecutive sequences of frames from the start to the end of the video. Therefore, the framework acquires the tracking output about these frame sequences and processes it, building the knowledge base and running the reasoning process. In this way, the proposed framework is able to work on subsets of frame sequences at a time and returns the inferred new axioms within seconds. This feature reduces reasoning computation time and makes the whole system suitable for real-time applications (e.g., real-time video streaming).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS


```
@prefix trackpoi: <http://www.semanticweb.org/danilo/ontologies
      /2016/1/ .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix georss: <http://www.georss.org/georss/> .

# POI_1 triples
TrackPOI:POI_1 rdf:type owl:NamedIndividual ,
TrackPOI:POI_1 rdf:type TrackPOI:POI .
TrackPOI:POI_1 rdf:type TrackPOI:Route .
TrackPOI:POI_1 TrackPOI:Administrative_area_level_1 "Campania" .
TrackPOI:POI_1 TrackPOI:Administrative_area_level_2 "Provincia di
      Salerno" .
TrackPOI:POI_1 TrackPOI:Administrative_area_level_3 "Fisciano" .
TrackPOI:POI_1 TrackPOI:Country "Italy" .
TrackPOI:POI_1 TrackPOI:Postal_code "84084" .
TrackPOI:POI_1 TrackPOI:Street_address "Anello Esterno, 84084
      Fisciano SA, Italy" .
TrackPOI:POI_1 georss:box "40.775762,14.787031,40.772937,14.785706"
      .
TrackPOI:POI_1 georss:point "40.7743843,14.7860267" .
TrackPOI:POI_1 TrackPOI:Name "Anello Esterno" .

# track_211_1395 triples
TrackPOI:Track_211_1395 rdf:type owl:NamedIndividual .
TrackPOI:Track_211_1395 rdf:type TrackPOI:Track .
TrackPOI:Track_211_1395 TrackPOI:hasRelationWith TrackPOI:POI_1 .
TrackPOI:Track_211_1395 TrackPOI:hasRelationWith TrackPOI:
      Track_239_1395 .
TrackPOI:Track_211_1395 TrackPOI:hasRelationWith TrackPOI:
      Track_252_1395 .
TrackPOI:Track_211_1395 TrackPOI:isNear TrackPOI:Track_252_1395 .
TrackPOI:Track_211_1395 TrackPOI:isInTheAreaOf TrackPOI:POI_1 .
TrackPOI:Track_211_1395 TrackPOI:hasHeight 58.309975411032354 .
TrackPOI:Track_211_1395 TrackPOI:hasSpeed 3.2 .
TrackPOI:Track_211_1395 TrackPOI:hasWidth 38.51992315031834 .
TrackPOI:Track_211_1395 georss:point "40.77454810242632,
      14.78530388911672" .
TrackPOI:Track_211_1395 TrackPOI:hasDirection TrackPOI:N .

# inferred triples for track_211_1395
TrackPOI:Track_211_1395 rdf:type <https://schema.org/Person> .
TrackPOI:Track_211_1395 TrackPOI:isInDangerOn TrackPOI:POI_1 .

# track_252_1395 triples
TrackPOI:Track_252_1395 rdf:type owl:NamedIndividual .
TrackPOI:Track_252_1395 rdf:type TrackPOI:Track .
TrackPOI:Track_252_1395 TrackPOI:hasRelationWith TrackPOI:POI_1 .
TrackPOI:Track_252_1395 TrackPOI:hasRelationWith TrackPOI:
      Track_211_1395 .
TrackPOI:Track_252_1395 TrackPOI:hasRelationWith TrackPOI:
      Track_239_1395 .
TrackPOI:Track_252_1395 TrackPOI:isInTheAreaOf TrackPOI:POI_1 .
TrackPOI:Track_252_1395 TrackPOI:isNear TrackPOI:Track_211_1395 .
TrackPOI:Track_252_1395 TrackPOI:hasHeight 50.35861512770976 .
TrackPOI:Track_252_1395 TrackPOI:hasSpeed 3.4 .
TrackPOI:Track_252_1395 TrackPOI:hasWidth 13.428964034055936 .
TrackPOI:Track_252_1395 georss:point "40.77412995839289,
      14.786379978047478" .
TrackPOI:Track_252_1395 TrackPOI:hasDirection TrackPOI:ENE .
# inferred triples for track_252_1395
TrackPOI:Track_252_1395 rdf:type <https://schema.org/Vehicle> .
TrackPOI:Track_252_1395 rdf:type TrackPOI:Car .
```

Listing 3. Assertional and inferred knowledge in Turtle statements for POI_1, :track_211_1395 and :track_252_1395. The inferred triples are in red.

A. Datasets

The dataset has been composed of recording live videos of some specific scenarios with a drone equipped with a camera. We have used the DJI F-450 drone, equipped with DJI NAZAM-V2 control unit, embedded GPS receiver and automatic mission. The on-board camera used is a Nilox F60 with 1028p full HD resolution and 175° wide view angle. We recorded videos because, at the best of our knowledge, there are no existing aerial view video datasets targeted at showing real critical or alerting situations, but usually they show only people or vehicles moving randomly in a scene. These videos[6] are recorded in the campus of the University of Salerno, specifically in an area which comprises a main route called Anello Esterno, heliport, parking, and some other places in the neighborhood like laboratories, department buildings, bars, etc. The drone mission has been planned with DJI Ground Station software, which provides the drone height and GPS data used by the framework in order to calculate object position, real dimensions, speed, as well as it retrieves the POIs appearing in the area. The first video shows two students walking together near the route until one goes away from the road and the other crosses the road while a car is coming. The second video captures the two students meeting once again, that are near and on a heliport while the helicopter is temporarily absent. videos 3 and 4 show a scenario similar to the one shown in video 1. These videos have been recorded between 3 and 4 P.M. in the summer. videos length is about a minute and half with a frame-rate of 25 frames/s. Each video has been divided into scenes of 15/22 s, time estimated for the scene evolution, where various events and objects appear. Additional videos, taken into account, are part of UAV123 dataset,[7] and present scenarios of people and vehicles moving in different environments such as route, roundabout, etc., even though their scenes do not present relevant alerting situations. videos are short (about 41 s), divided into three scenes, and present the same frame rate of 25 frames/s. Finally, the last two videos are retrieved from the Web and present more articulated situations, with some critical events. They seem interesting for the scenes evolution and the number of tracks appearing in the frame sequences. The experimentation is mainly based on the evaluation of the two main features: the capability of our system in the object identification and labeling and the recognition of critical or alerting situations during the scene evolution.

B. Object Identification

Correct identification of objects depends strongly on tracking results, because the recognition is partly based on bounding box dimensions. Tests do not involve the evaluation of the tracking algorithm (studied in [38]), but the capabilities of the reasoning process to identify the object type. To this purpose, a handmade ground truth of the video, has been extended with a semantic annotation, describing the object type. Each bounding box has been annotated with a label that is the TrackPOI ontology class which the object (in the bounding box) belongs to. Class individuals returned by our framework are compared to the "semantic" ground truth annotations at frame level. The classes involved in the object classification are all subclasses of *Track* class, mainly: *Person* and *Vehicle*; also Vehicle's subclasses *Car* and *Truck* (see Fig. 2) have been considered. The system evaluation in object identification is based on the precision and the recall measures. For each frame, all the tracks are retrieved with a SPARQL query, including the inferred statements about the class they belong to. According to the Pascal overlap criterion [39], if a calculated bounding box intersects with a bounding box in the ground truth for at

[6] https://drive.google.com/open?id=0B75yuWMeqbP5NVloZEIzc05jeW8
[7] https://ivul.kaust.edu.sa/Pages/Dataset-UAV123.aspx

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS

TABLE I
OBJECT LABELING TEST RESULTS

Video	Scenes	Frames	Precision	Recall	F1 score
1	1	578	0.94	0.72	0.82
	2	578	0.83	0.75	0.79
	3	578	0.88	0.83	0.85
	4	579	1.0	0.44	0.61
2	1	421	0.83	0.81	0.82
	2	421	0.95	1.0	0.97
	3	422	0.66	1.0	0.80
	4	422	0.89	1.0	0.94
3	1	506	1.0	1.0	1.0
	2	505	1.0	1.0	1.0
	3	506	1.0	0.78	0.88
	4	505	1.0	0.86	0.92
4	1	375	1.0	1.0	1.0
	2	375	1.0	1.0	1.0
	3	375	0.96	0.84	0.90
	4	374	1.0	0.89	0.94
5	1	340	1.0	0.85	0.92
	2	340	0.82	1.0	0.90
	3	340	1.0	0.82	0.90
6	1	343	1.0	0.73	0.84
	2	344	1.0	1.0	1.0
	3	344	1.0	1.0	1.0
7	1	342	1.0	1.0	1.0
	2	342	0.94	1.0	0.97
	3	343	1.0	0.91	0.95
8	1	345	1.0	0.91	0.95
	2	345	0.97	1.0	0.98
	3	345	1.0	1.0	1.0
9	1	610	1.0	0.76	0.86
	2	610	1.0	0.87	0.93
	3	610	1.0	0.89	0.94
10	1	610	1.0	0.7	0.82
	2	610	1.0	0.87	0.93
	3	610	1.0	0.83	0.91

most the 50%, then the two class labels are compared. With this evaluation method, the comparison holds when one of the following cases occur.

1) The two labels are equal: our system has correctly recognized the moving objects, i.e., a true positive is returned.
2) The two labels are not equal: our system individuates and identifies the object but does not recognize it correctly; this is a case of a false positive (FP).
3) The bounding box returned by our system does not intersect with a ground truth bounding box: that means the system does not recognize a true and effective object; it represents a false negative (FN).

Table I shows results on object labeling, displaying precision, recall, and F1-score for each scene in a video. Ten videos have been analyzed, split in several scenes; each one composed of several frames (see *Frames* in Table I), varying in a range from 340 to 610. The frame rate is 25 for all the videos except the last two, with a frame rate equal to 30. Let us remark that the framework does not evaluate results at each frame because there are very little changes both in objects and events, in a second. For this reason, our framework analyses from 15 to 23 frames per scene.

The system performance generally can be considered satisfactory. Table I shows that the recall value is greater than

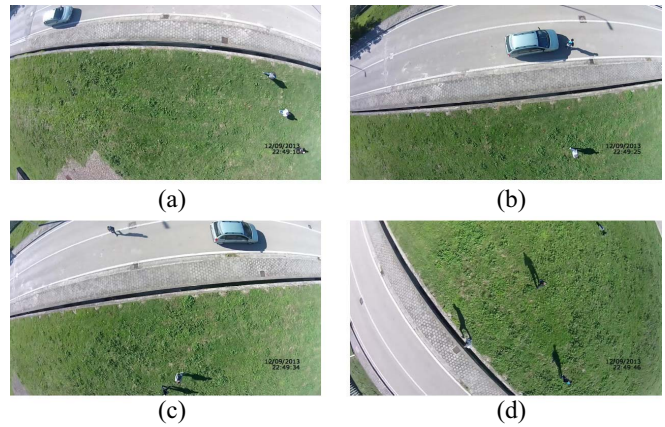

Fig. 5. Video 1, frames from scenes (a) 1, (b) 2, (c) 3, and (d) 4.

72% on all scenes of every video except for the first video, fourth scene, where the recall is 44%. Similar trend is observed for the precision, for each video scene. Fig. 7 shows the F1 score trend (and precision), by varying the recall. F1 curve has a strong growth with increasing recall starting from its lowest peak (0.61), due to low recall (0.44) on a scene, till to reach 1.0. On recall values greater than 0.7, there are just a few little variations due to some precision variations in value range 0.82–0.90. A precision decreasing also affects the F1 score curve, but the values are comprehensively high. The F1 trend reflects the good performance of the proposed system, whose capability at labeling objects improves with little improvements in tracking performance (growing recall values).

Looking at video 1, Fig. 5 shows four its own frames, taken respectively, from each scene. Video 1 recorded the two students walking close to the route, particularly, one of them crossing the road while a car is coming. In general, the system correctly recognizes the objects in scenes 1 and 3 [Fig. 5(a) and (c)]; and the precision for these scenes gets the highest values. Scene 2 [Fig. 5(b)] has some more FPs because the tracking algorithm returns some variable bounding boxes on the car and the student, when they are very close. The tracking algorithm does not detect the stopped car and generates some FNs. Scene 4 [Fig. 5(d)] presents the highest number of FN, because the tracking algorithm fails in the detection task, due to the ghost object presence as well as the environment element movements.

video 2 shows a different scenario: two students and other people walking and standing around to the area of the heliport. Four representative frames, each one taken from a different scene, are shown in Fig. 6. Also for this video, the system performances are good: the recall is always greater than 81% on all scenes, whereas the precision is generally high, except for the third scene where it reaches only 66%. In the first scene, three persons appear; the person walking to the center of the platform is always detected and correctly recognized, while the other two persons are not: when they are detected, their bounding boxes often have variable dimensions, with the resulting generation of possible FPs and FNs. Scene 2 shows the two students coming toward the center of the platform: they

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS

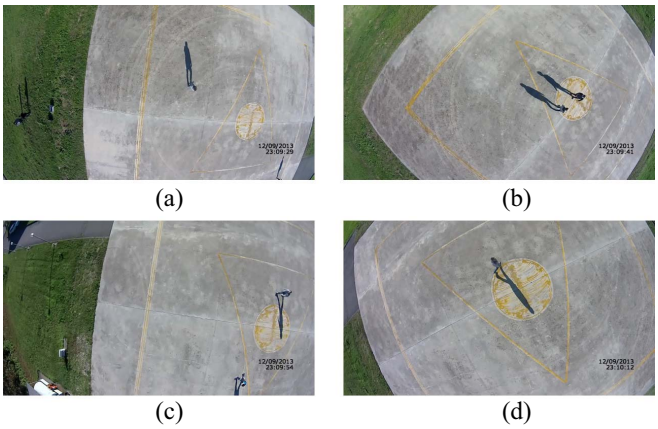

Fig. 6. Video 2, frames from scenes (a) 1, (b) 2, (c) 3, and (d) 4.

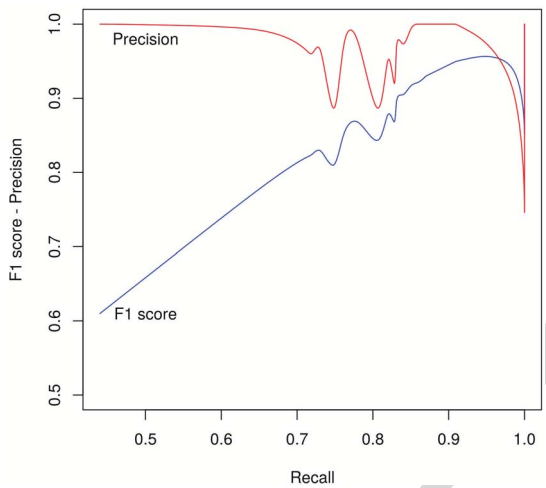

Fig. 7. Object labeling: F1-recall and precision–recall curves on the video collection.

are almost always correctly recognized except a very few FPs in some frames. Scene 3 presents the two students walking and stopping at the center of the heliport, in this case, the object labeling gets more FPs, due to excessive variable bounding boxes returned by the tracking algorithm. Finally, scene 4 shows students going back to the platform borders, and in this case, the precision rises to 89% due to the presence of a few FPs.

Videos 3 and 4 are so similar to video 1 in dynamics and contents, but the tracking algorithm returns a better output which is reflected in the results. In detail, in scenes 1 and 2 of video 3, the system correctly recognizes all the people moving toward the route, while in scenes 3 and 4 there are some FPs caused by the tracking algorithm output. For video 4, the system perfectly recognizes all objects in the first two scenes, while in scene 3, the tracking algorithm has some problems in detecting the student crossing the route when he passes near a car: this event badly affects the precision introducing with some FPs.

Also for videos 5–8, taken from UAV123 dataset, Table I presents good results on both precision and recall measures. Video 5, first scene shows a man walking on the grass, then he meets a second man (second scene) and finally in the third scene, the couple stops and has a chat. The system correctly recognizes the two men in every scene, yet it gets poor precision on the second scene, due to the presence of some FPs caused by the excessive speed of the object movements. Video 6 is set on a roundabout, the first and third scenes show just a few cars while the second shows more cars going in different directions. The framework recognizes perfectly all the vehicles in every scene, as it is evident in the table; only the recall for the first scene suffers the presence of some FNs. videos 7 and 8 show similar route and roundabout scenarios to those present in video 6, but taken from different illumination conditions and camera angles. Precision and recall results are quite similar to those of videos 5 and 6. Further tests have been carried out on UAV123 dataset, by considering a sample of 20 videos. They have been not reported in Table I because present similar scene and tracked objects, and consequently, similar results to videos 5–8.

Video 9 instead, captures an occurred car crash in the middle of a crossroad. This leads to additional critical situations: the crashed car forces the other vehicles to turn into the oncoming traffic, furthermore, road crossing pedestrians are obscured by the stopped vehicles. The video presents a great amount of moving objects appearing in the scene. The system resulting performance on this video are promising: precision values, in particular, are around 100% on all scenes. Recall mean value is around 80% with a minimum of 76%. Video 10 shows a scenario similar to video 9. In this video there are more cars and trucks, especially two long ones, that dangerously swerve into the oncoming traffic. The results evidence a very good precision: a great amount of objects are correctly labeled. Recall has a minimum of 70% on the first scene where some FNs occur, but on the other scenes gets very high values.

C. Situation Understanding

The system performances are also measured in terms of the situation detection, i.e., the individuation of critical events occurring in the scene and involving detected objects. Our system can detect events, by elaborating semantic data: it processes specific statements produced by the reasoning process (see Section V-F). These statements are rules or properties that can be activated when a critical/alerting situation (described by some statements) is recognized (if the system does not produce these statements, the scene is recognized as safe). Rules are designed indeed on events involving Google Maps POIs and moving objects, thus, when they are activated, new statements are generated, describing situations classified as critical.

In order to assess the capabilities of our system in detecting critical situations, a ground truth of the critical events has been manually built, and extended marking with annotation each critical event occurred in every video scene. In videos 1–4 (described in the previous section), events that produce critical situations are related to the road and the heliport. Specifically, the events involve tracks, mainly persons and vehicles and interact with both POIs. They are: 1) Ev1—walking or standing around the route or heliport and 2) Ev2—walking or standing near (more close to) the route or heliport and, when

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI:10.1109/TSMC.2017.2757462, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. SYSTEMS

the POI is the route, the events are: 1) Ev3—crossing/running the road when it is empty and 2) Ev4—crossing/running the road when a vehicle is coming (valid for both persons and vehicles), whereas for the heliport, they are: 1) Ev5—walking or standing in the area of the heliport near the borders and 2) Ev6—walking or standing in the area of the heliport at center.

videos 5–8 are set in route, park, and roundabout scenarios showing a single kind of object (vehicle or man), thus they present very few critical road-related events.

Events Ev1–Ev4 (i.e., except those related to the heliport) are also present in videos 9 and 10. Since these videos are set on a crossroad, events Ev3 and Ev4 more frequently are related to vehicles. In particular, Ev4 can refer to vehicle-person accidents, as well as only vehicle accidents. The system performance on discovering and identifying the critical situations has been evaluated by the accuracy measure, defined as the number of discovered (critical) events, detected per scene by our framework, with respect to the total number of events, defined by the annotated ground truth. More formally, given a scene s, the accuracy A_e on an event e occurring in the scene s, is the number of detected events S_e that match the total number of defined events E_e

$$A_e = \frac{S_e \cap E_e}{E_e}. \tag{1}$$

Table II shows test results on the situation recognition by considering the scenes from the videos, presented in the previous section. The accuracy A_e is evaluated for all the listed critical events. Events marked with "/" symbol do not occur or are not allowed for that specific video.

The experimental results are promising: critical events occurring in the videos are often detected mainly because of an efficient reasoning process and the high-quality data provided by the geometric tools, as well as the used map-based tools. These results are likely due to the nature of videos that are limited to two types of moving objects: 1) persons and 2) vehicles. Moreover, the reasoning model has been "tailored" to detect alerting situations, when people and vehicles come into play. Undiscovered critical situations may depend on various factors. First, an object that is not correctly retrieved for many frames in the scene can affect the detection of an alerting event. Then, a critical event is mainly related to object GPS coordinates, thus, if they are not exact, some positional relations (e.g., inTheAreaOf, inFrontOf, etc.) could be missing and consequently, a critical event could not be detected.

The accuracy for video 1 is satisfactory in all the scenes: in scene 1 [Fig. 5(a)], the two students coming near the route are correctly detected; in scene 2 [Fig. 5(b)] the system recognizes the two students and the third guy walking, respectively, near and around the route. In scene 3 [Fig. 5(c)], one student crossing the road twice is detected: the first time when a car is coming and the second time when the road is empty; the system also correctly detects the students while standing near the road. In scene 4, the system detects the students coming near the route, but a student walking on the sidewalk is detected as crossing the road, due to the low location accuracy of GPS Google Maps.

TABLE II
SCENE UNDERSTANDING TEST RESULTS

Video	Scenes	Accuracy (A_e)					
		Ev1	Ev2	Ev3	Ev4	Ev5	Ev6
1	1	1.0	1.0	/	/	/	/
	2	1.0	1.0	/	/	/	/
	3	/	1.0	1.0	1.0	/	/
	4	1.0	0.67	/	/	/	/
2	1	1.0	1.0	/	/	1.0	1.0
	2	/	1.0	/	/	1.0	1.0
	3	/	/	/	/	1.0	1.0
	4	/	/	/	/	1.0	1.0
3	1	1.0	1.0	/	/	/	/
	2	1.0	1.0	/	/	/	/
	3	1.0	1.0	/	/	/	/
	4	1.0	1.0	1.0	1.0	/	/
4	1	1.0	/	/	/	/	/
	2	1.0	/	/	1.0	/	/
	3	1.0	/	1.0	1.0	/	/
	4	1.0	0.82	1.0	/	/	/
6	1	/	/	/	1.0	/	/
	2	/	/	/	/	/	/
	3	/	/	/	/	/	/
7	1	/	/	/	1.0	/	/
	2	/	/	/	1.0	/	/
	3	/	/	/	/	/	/
8	1	/	/	/	1.0	/	/
	2	/	/	/	/	/	/
	3	/	/	/	1.0	/	/
9	1	1.0	1.0	/	1.0	/	/
	2	0.96	1.0	/	1.0	/	/
	3	1.0	1.0	/	1.0	/	/
10	1	0.92	0.87	1.0	1.0	/	/
	2	1.0	1.0	1.0	1.0	/	/
	3	1.0	1.0	1.0	1.0	/	/

Experimental results in video 2 confirm the good performance of the system in event identification. In scene 1 [Fig. 6(a)], a student coming toward the center of the heliport and other two persons standing near the heliport are correctly detected. In scenes 2 and 3 [Fig. 6(b) and (c)], the two students walk toward the center of the heliport and then stop for a while. The system still correctly recognizes the two events: walking on the border and on the center of the landing area. Both events are regarded as very critical because the students are inside the heliport area. Finally, in scene 4 [Fig. 6(d)], the system correctly identifies the other student coming to the border of the heliport.

Similarly to video 1, video 3 shows, in the first three scenes, a person walking on the lawn away from the route and the two students going toward the route; the fourth scene presents one of the two students crossing with and without a car coming. Also in these cases, the system successfully detects all the critical situations.

Video 4 also shows initially only a student walking on the lawn, then it shows another student crossing the route in scenes 2 and 3. In the final scene, there are the two students stopping near the road. Situations are correctly recognized except some Ev2 events in the fourth scene due to the low location accuracy of GPS Google Maps.

For videos 5–8, the system detects very few critical situations; these videos present just a single type of object: Person (video 5) or Vehicle (videos 6–8), moving in safe conditions

in a flat scenario with a very few elements. Video 5 indeed shows two people walking on a deserted lawn and doing nothing else. In fact, no event is detected (e.g., no result is presented in the table). Video 6 instead, shows some cars running around a roundabout: the only alerting situation occurs between two cars in the first scene, where a presumed collision (Ev4) was about to happen, and the system correctly recognizes it. Videos 7 and 8 present route events similar to those seen in video 6, with a consequent similar accuracy value.

Videos 9 and 10 are rich of events happening in the same time after a car crashing, with a car stopped in the middle of the crossroad. The results on video 9 are very good: in scene 1, the system correctly recognizes, as a critical situation, the occurrence of events Ev2 and Ev4, describing a person leaving the car and crossing while more cars are coming. Some other people walking near the road and crossing are also successfully detected (Ev1). In scene 2, the situation describing some cars going toward the stopped car is successfully recognized as critical, as well as some people walking near the road when cars are coming. In scene 3, all the events Ev4 are detected: they mainly refer to cars that invade the side of the oncoming traffic roadway. Finally, in video 10, scene 1, there are many different co-occurring events: people close to the road and crossing toward oncoming vehicles (Ev1, Ev2, and Ev4), cars and trucks that quickly drive into the oncoming traffic (Ev4). The system identifies the most of the occurring events, only few Ev1 and Ev2 events are lost.

VII. Conclusion

This paper introduces a novel approach to enhance the video tracking task, by adding semantic information to the tracks appearing in the video frames, in order to provide a high-level comprehensive description of the scenes recorded by the mobile camera video. The final goal is to get a more accurate object identification and labeling and, at the same time, to get critical situations and scene understanding based on the identified relations. The proposed framework exploits the semantic Web technologies to extend the assertional knowledge extracted by the video analysis. The reasoning model has been designed to infer new assertions that support the understanding of the dynamics in the scene evolution.

The experimentation in real applications shows interesting results on videos recorded with a camera-on-board drone. The add-on is the use of mobile camera, compared with the most works in literature that are based on fixed camera.

Future extensions will focus on improving the reliability of the system in object recognition and the dynamic estimation of the dangers affecting a specific scenario. The aim is providing an "alarm" measure, that allows identifying potential risks during the development of the scene: each identified scenario will be comprehensively described by a danger value which represents the alarming level detected in the scene.

References

[1] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Trans. Robot. Autom.*, vol. 19, no. 3, pp. 371–380, Jun. 2003.

[2] A. Symington, S. Waharte, S. Julier, and N. Trigoni, "Probabilistic target detection by camera-equipped UAVs," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Anchorage, AK, USA, May 2010, pp. 4076–4081.

[3] M. Ugliano, L. Bianchi, A. Bottino, and W. Allasia, "Automatically detecting changes and anomalies in unmanned aerial vehicle images," in *Proc. IEEE 1st Int. Forum Res. Technol. Soc. Ind. Leveraging Better Tomorrow (RTSI)*, Turin, Italy, Sep. 2015, pp. 484–489.

[4] M. Siam, R. ElSayed, and M. ElHelw, "On-board multiple target detection and tracking on camera-equipped aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Guangzhou, China, Dec. 2012, pp. 2399–2405.

[5] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 6, pp. 1019–1029, Jun. 2017.

[6] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[7] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid, "Unsupervised object discovery and tracking in video collections," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, 2015, pp. 3173–3181.

[8] F. D. Smedt, D. Hulens, and T. Goedemé, "On-board real-time tracking of pedestrians on a UAV," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Boston, MA, USA, Jun. 2015, pp. 1–8.

[9] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.

[10] C.-Y. Hsu, L.-W. Kang, and H.-Y. M. Liao, "Cross-camera vehicle tracking via affine invariant object matching for video forensics applications," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, San Jose, CA, USA, Jul. 2013, pp. 1–6.

[11] J. T. Lee, C.-C. Chen, and J. K. Aggarwal, "Recognizing human-vehicle interactions from aerial video without training," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Colorado Springs, CO, USA, Jun. 2011, pp. 53–60.

[12] J. Wang, J. Qian, and R. Ma, "Urban road information extraction from high resolution remotely sensed image based on semantic model," in *Proc. 21st Int. Conf. Geoinformatics*, Kaifeng, China, Jun. 2013, pp. 1–5.

[13] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 297–309, Feb. 2015.

[14] V. Bruni and D. Vitulano, "An improvement of kernel-based object tracking based on human perception," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 11, pp. 1474–1485, Nov. 2014.

[15] T. Chen and S. Lu, "Object-level motion detection from moving cameras," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2016.2587387.

[16] P. Huang, J. Cai, Z. Meng, Z. Hu, and D. Wang, "Novel method of monocular real-time feature point tracking for tethered space robots," *J. Aerosp. Eng.*, vol. 27, no. 6, 2013, Art. no. 04014039.

[17] L. Chen, P. Huang, J. Cai, Z. Meng, and Z. Liu, "A non-cooperative target grasping position prediction model for tethered space robot," *Aerosp. Sci. Technol.*, vol. 58, pp. 571–581, Nov. 2016.

[18] J. Cai, P. Huang, B. Zhang, and D. Wang, "A TSR visual servoing system based on a novel dynamic template matching method," *Sensors*, vol. 15, no. 12, pp. 32152–32167, 2015.

[19] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 4. The Hague, The Netherlands, Oct. 2004, pp. 3099–3104.

[20] R. Brehar, C. Fortuna, S. Bota, D. Mladenic, and S. Nedevschi, "Spatio-temporal reasoning for traffic scene understanding," in *Proc. IEEE Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Cluj-Napoca, Romania, Aug. 2011, pp. 377–384.

[21] M.-C. Chuang, J.-N. Hwang, J.-H. Ye, S.-C. Huang, and K. Williams, "Underwater fish tracking for moving cameras based on deformable multiple kernels," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 9, pp. 2467–2477, Sep. 2017.

[22] M. Eldib *et al.*, "A low resolution multi-camera system for person tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 378–382.

[23] X. Li and H. Lu, "Object tracking based on local learning," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Sep. 2012, pp. 413–416.

[24] W. Jiang, C. Xiao, H. Jin, S. Zhu, and Z. Lu, "Vehicle tracking with non-overlapping views for multi-camera surveillance system," in *Proc. IEEE 10th Int. Conf. High Perform. Comput. Commun. IEEE Int. Conf. Embedded Ubiquitous Comput.*, Nov. 2013, pp. 1213–1220.

[25] Z. Zhang and F. Cohen, "3D pedestrian tracking based on overhead cameras," in *Proc. 7th Int. Conf. Distrib. Smart Cameras (ICDSC)*, Palm Springs, CA, USA, Oct. 2013, pp. 1–6.

[26] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.

[27] C. Shen, N. Yu, W. Li, and W. Zhou, "Moving object detection of dynamic scenes using spatio-temporal context and background modeling," in *Proc. 6th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Hefei, China, Oct. 2014, pp. 1–4.

[28] C. F. Crispim-Junior *et al.*, "Semantic event fusion of different visual modality concepts for activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1598–1611, Aug. 2016.

[29] C. Kothari, J. Qualls, and D. Russomanno, "An ontology-based data fusion framework for profiling sensors," in *Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT)*, Indianapolis, IN, USA, May 2012, pp. 1–6.

[30] J. Gomez-Romero, M. A. Patricio, J. Garcia, and J. M. Molina, "Context-based reasoning using ontologies to adapt visual tracking in surveillance," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveillance (AVSS)*, Genoa, Italy, Sep. 2009, pp. 226–231.

[31] M. Andersson *et al.*, "Activity recognition and localization on a truck parking lot," in *Proc. 10th IEEE Int. Conf. Adv. Video Signal Based Surveillance (AVSS)*, Kraków, Poland, Aug. 2013, pp. 263–269.

[32] X. Zhang *et al.*, "Human pose estimation and tracking via parsing a tree structure based human model," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 5, pp. 580–592, May 2014.

[33] J. Garcia, A. Gardel, I. Bravo, J. L. Lázaro, and M. Martinez, "Tracking people motion based on extended condensation algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 3, pp. 606–618, May 2013.

[34] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V.-K. Papastathis, and M. G. Strintzis, "Knowledge-assisted semantic video object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1210–1224, Oct. 2005.

[35] H. He and B. Upcroft, "Nonparametric semantic segmentation for 3D street scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 2013, pp. 3697–3703.

[36] Y. Li, Y. Guo, Y. Kao, and R. He, "Image piece learning for weakly supervised semantic segmentation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 648–659, Apr. 2017.

[37] D. Cavaliere, S. Senatore, M. Vento, and V. Loia, "Towards semantic context-aware drones for aerial scenes understanding," in *Proc. 13th IEEE Int. Conf. Adv. Video Signal Based Surveillance (AVSS)*, Colorado Springs, CO, USA, Aug. 2016, pp. 115–121.

[38] A. Greco, A. Saggese, G. Tozzi, and M. Vento, "Detection and tracking of moving objects by flying cameras using an adaptive test and set approach," in *Proc. 22nd Comput. Vis. Win. Workshop (CVWW)*, 2017, pp. 1–6.

[39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

Danilo Cavaliere received the master's degree in computer science from the University of Salerno, Fisciano, Italy, in 2014, where he is currently pursuing the Ph.D. degree.

Since 2015, he has been a Pre-Doctoral Fellow with the Computer Science Department and the Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno. His current research interests include artificial and computational intelligence, machine learning, data mining, and knowledge discovery, areas in which he has published some papers.

Vincenzo Loia (SM'08) received the B.S. degree from the University of Salerno, Fisciano, Italy, in 1985, and the M.S. and Ph.D. degrees from the University of Paris VI, Paris, France, in 1987 and 1989, respectively, all in computer science.

Since 1989, he has been a Faculty Member with the University of Salerno, where he teaches safe systems, situational awareness, IT project and service management and is currently the Chair and a Professor of computer science with the Department of Management and Innovation Systems. His current research interests include soft computing, agent technology for technologically complex environments Web intelligence, and situational awareness. He was a principal investigator in industrial research and development projects and in academic research projects. He has authored over 400 original research papers in international journals, book chapters, and international conference proceedings.

Dr. Loia is the Coeditor-in-Chief of *Soft Computing* (Springer) and the Editor-in-Chief of *Ambient Intelligence and Humanized Computing* (Springer). He is an Associate Editor of various journals, including the IEEE TRANSACTIONS ON SYSTEM, MAN AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON FUZZY SYSTEMS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS. He holds several roles in the IEEE Society in particular for Computational Intelligence Society (the Chair of the Emergent Technologies Technical Committee, IEEE CIS European Representative, and the Vice-Chair of the Intelligent Systems Applications Technical Committee).

Alessia Saggese (M'14) received the Ph.D. degree in computer engineering from the University of Salerno, Fisciano, Italy, and the University of Caen Normandy, Caen, France, in 2014.

She is currently an Assistant Professor with the University of Salerno. Her current research interests include basic methodologies and applications in computer vision and pattern recognition.

Dr. Saggese has been a member of the International Association for Pattern Recognition Technical Committee 15 on Graph-Based Representations in Pattern Recognition since 2012.

Sabrina Senatore (M'06) received the M.S. and Ph.D. degrees in computer science from the University of Salerno, Fisciano, Italy, in 1999 and 2004, respectively.

Since 2005, she has been a Faculty Member with the University of Salerno, where she is an Associate Professor of computer science. Her current research interests include the development and application of intelligent systems based on the combination of techniques from soft computing, computational intelligence, text mining, Web information retrieval, semantic Web, and machine learning, and software agents, areas in which she has published numerous papers.

Dr. Senatore is an Editorial Board Member of the *Applied Intelligence Journal* and the *International Journal of Computational Intelligence Theory and Practice*.

Mario Vento (M'14) received the Ph.D. degree in computer engineering from the University of Napoli Federico II, Naples, Italy, in 1989.

He is currently a Full Professor of computer science and artificial intelligence with the University of Salerno, Fisciano, Italy, where he is the Coordinator of the Artificial Vision Laboratory and the Dean of the Department of Information and Electrical Engineering and Applied Mathematics. His current research interests include real-time video analysis and interpretation for video surveillance applications, classification techniques, either statistical, syntactic, and structural, exact and inexact graph matching, multiexpert classification, and learning methodologies for structural descriptions.

Prof. Vento served as the Chairman of the International Association Pattern Recognition (IAPR) Technical Committee 15 on Graph-Based Representation in Pattern Recognition from 2002 to 2006. He is a Fellow Scientist of the IAPR.