A human-like description of scene events for a proper UAV-based video content analysis

Danilo Cavaliere[a], Vincenzo Loia[b,*], Alessia Saggese[a], Sabrina Senatore[a], Mario Vento[a]

[a] *Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata - DIEM*
Universitá degli Studi di Salerno - Fisciano, Italy
[b] *Dipartimento di Scienze Aziendali - Management & Innovation Systems - DISA-MIS*
Universitá degli Studi di Salerno - Fisciano, Italy

Abstract

In Video Surveillance age, the monitoring activity, especially from unmanned vehicles, needs some degree of autonomy in the scenario interpretation. Video Analysis tasks are crucial for the target tracking and recognition; anyway, it would be desirable if a further level of understanding could provide a comprehensive, high-level scene description, by reflecting that human cognitive capability of providing a concise scene description that comes from the analysis of involved objects relationships and actions.
This paper presents a smart system to identify mobile scene objects, such as people, vehicles, automatically, by analyzing the videos acquired by flying drones, along with the activities they carried out, so as to depict what it happens in the scene from a high-level perspective. The system uses Artificial Vision methods to detect and track the mobile objects and the area where they move, and Semantic Web technologies to provide a high-level description of the scenario. Spatio/temporal relations among the tracked objects as well as simple object activities (events) are described. By semantic reasoning, the system is able to connect the simple activities into more complex activities, that better reflect a human-like description of a scenario portion. Tests conducted on several videos, showing scenarios set in different environments, return convincing results which affirm the effectiveness of the proposed approach.

Keywords: Activity Detection, Semantic Web technologies, Activity composition, Object Classification, Video Tracking, OWL

1. Introduction

A report conducted by Information Handling Services (IHS) [1] about Top Video Surveillance Trends in 2016 confirms that one of the big trends in surveil-

*Corresponding author
Email address: loia@unisa.it (Vincenzo Loia)

lance applications is related to mobile cameras. About 66 millions of cameras have been installed only in 2016, and a growing interest has been devoted to *mobile* cameras, namely to those cameras though for and installed over mobile platforms such as unmanned vehicles or body-worn. Indeed, about 2% of cameras installed in 2016 (1.3 million cameras) are mobile. In the last decade, cameras mounted over unmanned vehicles, both Unmanned Ground Vehicle (UGV) and Unmanned Aerial Vehicle (UAV) attracted scientific communities working in the computer vision and artificial intelligence fields. Such cameras allow to potentially acquire any scene where some event of interest occurs (even if no fixed cameras are already available), thanks to the possibility to remotely pilot the vehicles over the desired place.

In particular, cameras mounted over unmanned vehicles, both Unmanned Ground Vehicle (UGV) and Unmanned Aerial Vehicle (UAV) attracted in the last decade scientific communities working in the computer vision and artificial intelligence field. Such cameras allow to potentially acquiring any scene where some event of interest occurs (even if no fixed cameras are already available), thanks to the possibility to remotely pilot the vehicles over the desired place. The UAV drones indeed, are having a strong impact on the public safety market.t.

In a 2016 report [2], Goldman Sachs estimated drone technologies will reach a total market size of 100 billion between 2016 and 2020: about 70% will be related to military activities, 17% to the consumer market and the remaining 13% for commercial business, that could reach something like 13 billion between 2016 and 2020. Furthermore, IHS report [1] confirms that a number of police forces decided to extend their surveillance systems to the drones.

Anyway, the large number of mobile cameras *flying* in the sky and thus of videos now available is more and more requiring the designing of algorithm able to automatically identify the movement of the objects populating the scene (the so-called multi-objects tracking algorithms) [3, 4]. At the same time, they should automatically analyze such trajectory so as to understand what it is happening in the scene and if something potentially dangerous may occur, so as to immediately alerting the operators in charge of the security.

The growing reliability achieved in the last years by multi-camera tracking algorithms [5, 6, 7] has moved the attention towards those algorithms able to analyze and interpret moving objects behaviors automatically.

Within this context, this work proposes a novel hybrid framework which builds high-level knowledge by integrating tracking and object classification data, along with semantic, contextual information targeted at interpreting complex object activities occurring in the video. The system interprets a video stream by providing a high-level scene description, through the event/action identification of the (moving) objects, populating the scenario and their interactions with the environment and the other (moving or fixed) objects. The goal is recognizing dynamic aspects in evolving scenarios by combining simple activities carried out by the scene objects into complex activities, to provide a high-level abstraction scene and enhance the overall situation understanding.
In detail, the contribution of this work is manifold. The system aims at achieving:

- a frame-by-frame object classifier to determine the actual identity for each tracked object.
- An area recognition to distinguish the kind of environment where the object is moving.
- A scene ontology to encode high-level spatio/temporal relations among scene objects (i.e., people, vehicles, etc), and between scene objects and the environment.
- A rule-based model to discover simple activities from the detected spatio/temporal relations at each time instant of the video.
- A high-level description of the main activities occurring in the evolving scenario by activity composition connected by the timeline.

The paper is structured as follows. Section 2 presents an overview on activity recognition systems. Section 3 introduces the whole system and the interaction of the main components. Section 4 is devoted to present the modules for scene object and environmental feature recognition; Sections 5 and 6 describe the generation of high-level knowledge, and the semantic enhancement to detect simple and complex activities, respectively. A case study on a road scenario is described in Section 7, whereas Section 8 shows the system performance, presenting the experimental results that validate the effectiveness of the system components. Conclusion closes the paper.

2. Related Work

Current trends in the Video Surveillance field evidence the main role of intelligent systems in acquiring and understanding scenarios. A UAV is considered "smart" if it is equipped with a semantic-based reasoning component, enabling it to capture heterogeneous information on the scene and then, reasoning about events and activities, occurring in the environments, in order to get an overall scene understanding. To this purpose, the following sections introduce a review of recent literature on the intelligent systems and the use of high-level knowledge to support activity detection.

2.1. Intelligent systems

A UAV to perform scenario detection is as highly desirable as complex to achieve in the surveillance and monitoring systems.
UAV movements bring some issues to scenario interpretation from a high-level perspective. UAV can fly over different environments in a few of seconds, this causes the loss of reference points in the scene. The loss of reference points complicates the recognition of object action and interaction with the environmental elements of the scene [8]. Moreover, the ever-changing outside scenarios, caught on camera by the UAV, make even more difficult the interpretation of events occurring in the video scenario. Scenario interpretation requires the

understanding of heterogeneous environments. To this purpose, the Machine Learning methodologies alone are not enough to support scenario interpretation, because they need high amounts of samples to be trained [9, 10], and do not possess cognitive capabilities to allow a deeper understanding of the object actions and scene events.

In order to achieve high-level scenario comprehension, intelligent systems are often taken into consideration. These systems emulate cognitive reasoning by employing an ontology, representing high-level knowledge on a domain. Reasoning over a scene ontology, representing knowledge on the video scene, can support the deduction of new facts on the scenario [11]. Some solutions proposed in literature focus on data fusion, collecting information from heterogeneous sources [12]. Some approaches are aimed at generating high-level contextual knowledge to improve scenario interpretation through contextual reasoning, and help decision-makers to deal with sensor imprecision [13]. Some solutions are designed for specific environments, so that they present ad-hoc scene ontologies [14], generally exploiting scene segmentation, to represent environment areas and allow deduction of events and object activities [15]. Obviously, ontologies built on specific applications are not reusable for other environments caught on UAV camera. In order to build more adaptable ontologies, some trends include spatial [16] and temporal information [17] to describe the events occurred in the scenario. Some approaches [18], [16] specialise ontologies and query to model places at different levels of granularity (i.e. states, regions, cities) to detect place areas. Our approach, instead, detects place areas by using an area classifier and retrieves additional information on the environment from external sources, exploiting databases and geo-positional map services, such as Google Maps. This information allows to model knowledge about different kinds of outside environments.

The proposed approach introduces a new way to build a human-like description of the observed scenario as composed of high-level activities by starting from a video stream. Contrary to approaches stating a simple message or reporting raw data, our framework codes and generates high-level knowledge and returns a refined set of people or vehicle actions detailing what happened in the observed UAV video.

2.2. Activity detection

Recent trends are aimed at building scene ontologies to elicit knowledge about events and activities carried out by the scene objects [15]. Generally, these models are thought to deal with one well-known domain, kind of environment and application (i.e. activity daily living), so that these approaches exploit a priori knowledge to build the scene ontology [19]. UAVs could fly over different kinds of environments and catch different kinds of objects and situations. So, a priori knowledge [20], or pose classification [21], could not be reliable, available or enough to detect activities. The desirable thing is to build models suited to accomplish activity detection in different heterogeneous environments. Our solution provides an adaptable model to detect people and vehicle activities in different contexts.

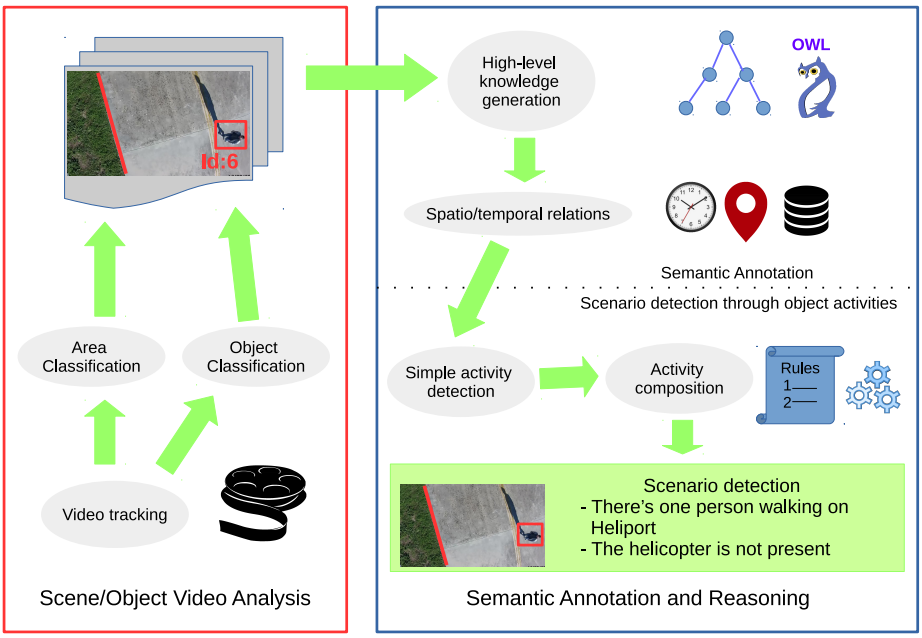

Figure 1: Framework overview

Other solutions in literature enhance the scene ontology with knowledge about space and time. Generally, these approaches employ fixed-sized temporal windows to detect events through the analysis of video time intervals, and store the most relevant detected events [22, 23]. These solutions find other issues related to the window management, correct size choice and evaluation of relevant activities, that could happen at same time or in distinct time intervals throughout the video [24]. Our model, instead, firstly defines spatio/temporal relations among the scene objects, and between the scene objects and the environment. Then, it contextualizes these relations with knowledge on objects to detect simple activities. Higher-level activities are then deduced through simple activity composition. Unlike the most trends in literature, aimed at directly detecting activities by exploiting patterns, our approach not only detects activities, but also introduces a higher-level incremental activity modelling that allows to better contextualize the activities over time and achieve higher-level abstraction and a better comprehension of the scene.

3. Overview

Figure 1 shows the logical overview of the system, evidencing two macro-area: *Scene/Object Video Analysis*, and *Semantic Annotation and Reasoning*. These are the main components of the system, that are in charge of the object recognition in the scenario (through video tracking and classification algorithms) and the

semantic annotation of objects (through semantic web technologies), respectively. The input data is a video recorded by a flying UAV. The *Scene/Object Video Analysis* component accomplishes the tracking algorithm on the video to object detection and recognition through frames. As shown in Figure 1, this component is in turn composed of three modules: each one achieves a specific processing on the input video. After the *Video tracking* task, the *Object Classification* accomplishes an object classification task, identifying and labeling the objects appearing in the video; the *Area Classification* module instead, detects area contours of distinct places in the environment (e.g. roads, grass, etc.).
The *Semantic Annotation and Reasoning* component aims at the semantic enrichment of scenario: it collects the data processed by the *Scene/Object Video Analysis* component and produces statements describing the scenario and involved objects at a semantic level. Specifically, the *High-level knowledge generation* module generates semantic annotations on object identity and place areas. It uses an ad-hoc ontology designed to describe scenarios populated by moving and fixed scene objects. Relations and interactions between objects and environment are processed by *Spatio/temporal relations* module that codes spatial basic relations among the scene objects in semantic statements. Finally, the remaining components are in charge of the knowledge inference on the scene by relating all the object activities occurring in a spatio/temporal context. *Simple activity detection* module detects general object activities by relating the object identity to tracking data and spatio/temporal relations at each time instant, then *Activity composition* module composes simple activities over time in order to deduce more articulated and specialised activities for each object. Activity composition acts to put the detected activities of the involved scene objects in the right context, with respect to time, space and the environment. *Scenario detection* module collects the revealed activities to provide a human-like description of the occurred scenario.

4. Scene/Object Video Analysis

This module automatically analyses the sequence of images acquired by the camera so as to answer to the following three questions:

1. Are there any objects moving in the scene?
2. What is the typology of objects moving in the scene?
3. What is the category of the scene where the objects are moving?

The answer to each question is reported in the following. In particular, the tracking algorithm used for identifying the objects moving in the scene (question 1) is detailed in Subsection 4.1; the algorithm proposed for identifying the typology of the objects moving in the scene (question 2) is detailed in Subsection 4.2; finally, the approach considered for understanding the typology of the scene where the object is moving (question 3) is detailed in Subsection 4.3.

An overview of the interaction between such components is reported in Figure 1, on the left side of the image (*Scene/Object Video Analysis* module).

4.1. Detection and Multi-target tracking

The aim of the tracking algorithm is to automatically analyze the sequence of images acquired by the camera mounted on board of the drone so as to extract the set of moving object trajectory. In more details, given the current frame F_t and the set of objects O_{t-1} detected until the previous frame, the tracking algorithm aims at updating the trajectories of the objects $O_t = \{O_t^1, ..., O_t^{|O|}\}$, drawing them frame by frame. In this paper, we adopt the detection and tracking algorithm we recently proposed in [6]. The detection algorithm aims at identifying, frame by frame, the so called blobs, namely those objects moving in the scene at the current frame. Considering that the camera is moving, it is not possible to exploit any background updating and subtraction algorithm [25]. Thus, in order to separate the ego motion (namely the motion of the camera moving on board of the flying drone) from the movement of the objects, we exploit preliminary camera compensation algorithm, aiming at estimating the direction and the magnitude of the camera movement between two consecutive frames. Then, we combine two different approaches, based on the foreground mask extraction and on the extraction of some salient points, respectively.

Given the sets of detected blobs, the tracking aims at performing the best possible association between blob(s) and the corresponding object(s), tracked until the previous frame. Note that the association is not always $1:1$ (one object with one blob); indeed, the following situations need to be managed, namely the splits and the merges. In the first situation an object is broken into two or more blobs due to an error during the detection step, thus one object needs to be associated with more than one blob $(1:n)$. Viceversa, in the second situation two or more objects merge in a single blob $(n:1)$; note that, due to the nature of the tracking problem from flying cameras (having a top-down view), the only merge that can happen is related to objects very close each other, and not to occlusions among moving objects or occlusions between a moving object and a background object (for instance a person partially behind a wall or a pole). In order to manage with the above mentioned issues, we combine a traditional forward chain based on data association with a backward chain: a local data association (between blobs and objects) is performed during the forward chain and the reliability in such association is evaluated. In case the reliability is not sufficiently high (for instance due to a split or to a merge), then the backward chain is activated, the operating parameters are accordingly adjusted and the forward chain is activated again with the new operating parameters. In other words, the forward chain is performed starting from a startup settings and is iteratively repeated with refined settings (automatically generated by the backward chain) until the confidence in the local data association is high enough. An overview of the architecture of the detection and tracking algorithm is reported in Figure 2.

Given the centroid of the object at the current frame (x_t^i, y_t^i) and at the previous frame (x_{t-1}^i, y_{t-1}^i), its direction d_t^i is also computed. In more details, we consider the eight directions shown in Figure 3, namely North (N), South (S), West (W), East (E) and the four combined directions, namely NE, SE, NW,

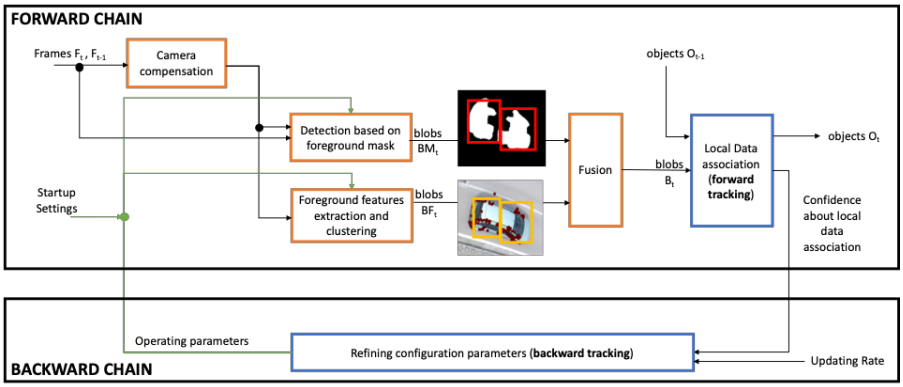

Figure 2: Architecture of the detection and tracking algorithm.

SW.

Then, for each object, the directions computed in the last N frames are considered and the one with the highest occurrence is assigned to the object.

4.2. Object classification

Once updated its trajectory, each object is classified in one of the following three classes: *person* (P), *vehicle* (V) and *unknown* (U). A two steps evaluation is considered: (1) the classification is performed frame by frame and then (2) a majority voting approach is employed, given all the classes associated frame by frame to each object. In practice, the class of the objects is computed in real time (frame by frame) and a decision is taken frame by frame by considering all the occurrences of that object up to the current frame. As evident, the higher is the number of occurrences of an object, the higher will be the reliability in the classification.

As for the frame by frame classification, we observe that the shape is a good property for representing the objects: indeed, the vehicles have a regular shape, while the persons have an irregular shape, due to their nature of non rigid objects. Starting from this consideration, we decide to represent each object by means of an HOG descriptor, which provides a measure about the shape. For each occurrence of the object O_t^i, its class at the current frame c_t^i is thus computed by employing a linear multiclass SVM, based on one-versus-all strategy.

Note that, for each object, a set of classes is provided; let's introduce the i-th object O^i and its occurrences until the frame t:

$$O^i = \{O^i_{t-|T|}, ..., O^i_t\}, \tag{1}$$

being $|T|$ the number of frames in which O^i appeared up to now, and then $t - |T|$ the first frame in which the object is visible inside the scene. For each occurrence of the object, a decision about the class to which the object belongs to is provided; it implies that at the t-th frame, $|T|$ potential classes for that

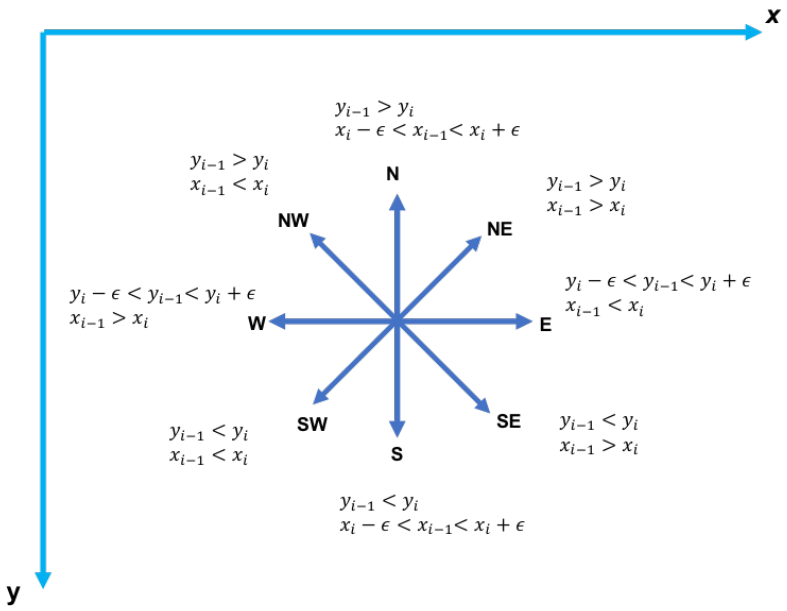

Figure 3: Quantisation of the movement of the objects in eight directions.

objects are available:

$$\{c^i_{t-|T|}, ..., c^i_t\}. \tag{2}$$

Finally, a majority voting approach is introduced; each decision is considered as a vote for a class, and the class $c_j \in \{\text{P,V,U}\}$ with the highest number of votes, c^i, is the winner for the object O^i:

$$c^i = \underset{j \in \{1,...,3\}}{\text{argmax}} \left(\sum_{k=1}^{|T|} h_{kj} \right) \tag{3}$$

where

$$h_{kj} = \begin{cases} 1 & if \quad c^i_k = c_j \\ 0, & otherwise \end{cases} \tag{4}$$

4.3. Area categorization

In order to understand the scene, and in particular the typology of the area where the object is moving, we introduce an algorithm for classifying the portions of the scene in the following three classes: *street*, *grass* and *unknown*. In more details, the scene is partitioned into non overlapped sliding patches, and each patch is classified by combining two complementary information: LBP features, able to take into account the texture of the area to be analysed, and histogram color, able to take into account color information. The two feature vectors are

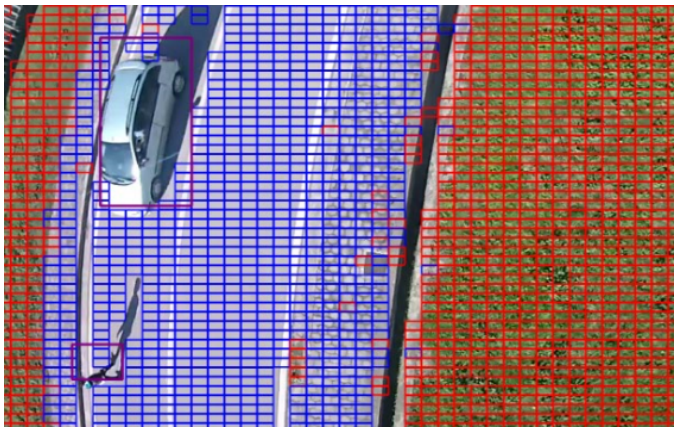

Figure 4: Example of area categorization. Blu patches are classified as road, while red patchesas grass.

fused and a K-NN classifier is used. An example is shown in Figure 4, where blue rectangles identify patches classified as road, while red rectangles correspond to patches classified as grass.

In order to provide to the *Semantic Annotation and Reasoning* module the information about the region instead than a single patch, each patch is considered as a pixel and morphological operators are applied; in particular, erosion and dilation are used so as to remove outliers patches, namely those patches (wrongly) classified as belonging to the class c_i and whose neighbouring patches belong to c_j. Finally, connected components are found, so that each region can be identified by a polygon.

5. High-level knowledge generation: ontology modeling and population

After the video analysis activities, achieved by *Scene/Object Video Analysis*, the data flow passes to *Semantic Annotation and Reasoning* that generates high-level knowledge on the whole scenario present in the video. The TrackPOI ontology [11] is used to describe the scenario at a semantic level: it is designed for modeling road scenario. The ontology schema specifies two main entities: the mobile and fixed objects. The mobile objects are the main actors of the scene; they could be people, animals, vehicles or other moving objects carried or pushed by living beings. Formally, $\hat{M} = \{\hat{o}_1, \hat{o}_2, ..\}$ is the set of mobile objects, while $F = \{y_1, y_2, ...\}$ is the set of the fixed objects of the scene, that are composed by environmental static features, identifying more or less extended places generally present in outside scenarios such as parks, roads, parking lots, as well as stores, ATMs, etc.

Each mobile object $\hat{o}_i \in \hat{M}$ is a sequence of tracks $\hat{o}_i = \{\hat{o}_i^{t_1}, \hat{o}_i^{t_2}, ..., \hat{o}_i^{t_n}\}$, where each track $\hat{o}_i^{t_j}$ represents the mobile object in a specific time instant t_j, with

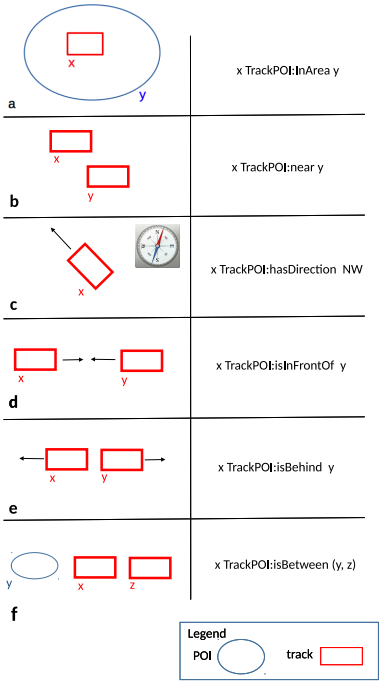

Figure 5: TrackPOI relations

$j = 1, \ldots, n$ of the video. The tracks of each object $\hat{o}_i$ and the fixed objects in the F set are, respectively, directly coded into two correspondent TrackPOI ontology classes. Tracks are represented by instances of the *Track* class, which has subclasses modeling two specialized types of track: Vehicle and Person. The class, which the instance belongs to, is identified according to the object classification output. Thus, if classification recognizes a tracked object as person, an instance of *Person* will be generated and added to the knowledge base. Otherwise, the object recognized as Vehicle will trigger the generation of a *Vehicle* class instance. The fixed objects are detected by using Google Maps service [11], which provides data about the identity and features of the Points of Interest (POIs), and possible places appearing in the scene. TrackPOI ontology models these entities as instances of the POI class.

The ontology has been extended with spatial relations among the tracks, and between the tracks and the POIs. For the sake of completeness, the whole list of relations is shown in Figure 5 (for additional details, see [11]). For example, properties such as *inArea* and *nearestPlace* describe relations between a track and the place where it appears. Assertions with these properties are generated according to the area classifier results. The spatial relations are asserted for the track in each video frame; the frame number is associated with the discovered relations as well as the frame instant where they happen, in order to get a complete description of track relations both in terms of space and time.

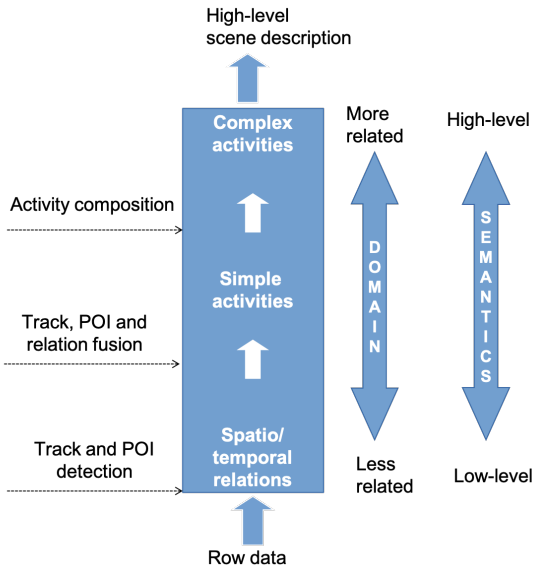

Figure 6: Activity modeling: from the row data to the high level scene description, by an incremental definition of the activities

6. Context and activity-based situation understanding

Once the knowledge base has been populated with tracks and POIs, and the spatial/temporal relations have been defined, the *Scenario detection through object activities* subcomponent may take charge of activity detection. The complex activity detection is achieved in an incremental way: the idea behind this approach is identifying activities or events that involve mobile objects and then composing these activities in more complex and high-level activities. Figure 6 describes the incremental abstraction model composed of different levels of knowledge achieved by performing several steps: spatio/temporal relations are built on the detected tracks and POIs (*Track and POI detection* step). The built spatio/temporal track relations are fused with environmental information on tracks and POIs to detect simple activities (*Track, POI and relation fusion* step). Then, simple activities are connected and composed with respect to space, time and environment where they appear, to describe complex activities expressing higher-level knowledge on the observed scene (*Activity composition* step).

Figure 6 shows a growing level of semantics, starting from the simple spatio/temporal relations to get a human-like description of complex activities; at the same time, the domain-dependence increases: as the activities becoming complex, so they are more specialized.

6.1. Simple activity detection

The spatio/temporal relations designed in the TrackPOI ontology represent elementary general activities where a track can be involved. So, for instance, the

Trackpoi:inArea relation represents the static elementary activity of standing in the area of a specific POI. The relation associates the track, at the instant t, with the spatial data (i.e., POI, pixel data) and video time. The reasoning model can enhance the knowledge base by inferring new statements over these ontological relations. As stated, the track spatio/temporal relations can be merged with other collected track data (i.e., dimensions, speed, direction) and along with the involved POI, allow the detection of higher-level activities. Let us remark that these activities are labeled "simple" because they are detected by directly fusing the spatio/temporal relations with the knowledge about the track and POI involved in the relation. Simple activities can be considered as binary relations between the track performing the activity and the object or place of the activity. Recalling the previous definitions of mobile object and fixed object sets, provided in Section 5, the simple activity is defined as follows:

Definition 1. ***Simple activity.*** *Let $F = \{y_1, y_2, ..., y_p\}$ be the fixed object set and $\hat{o}_i$, $\hat{o}_j \in \hat{M}$ be distinct mobile objects, each one composed of tracks at distinct time instants $\hat{o}_i = \{\hat{o}_i^{t_1}, \hat{o}_i^{t_2}, ..., \hat{o}_i^{t_n}\}$, $\hat{o}_j = \{\hat{o}_j^{t_1}, \hat{o}_j^{t_2}, ..., \hat{o}_j^{t_n}\}$. A simple activity S_t carried out by the mobile object $\hat{o}_i$, at a time instant t, is expressed as the binary relation R between the track $\hat{o}_i^t$ of the mobile object $\hat{o}_i$ and some object z:*

$$S_t = \ < R, \hat{o}_i^t, z >_t \tag{5}$$

$$where\ z = \begin{cases} \hat{o}_j^t, & \hat{o}_j^t \in \hat{o}_j, \quad with \ \ j \neq i \\ y_h, & y_h \in F, \quad with \ \ 1 \leq h \leq p \end{cases}$$

These simple activities are also more contextualized than the simple spatio/temporal relations, although they are still quite general and capable of happening in many different scenarios (i.e., going towards someplace, accelerating, decelerating, etc.).

As an example of a simple activity, let us consider a video showing a car running on a road. To detect a car moving on a road, the spatio/temporal relations, stating that the car is on a road at the instant t, must be combined with the context-based features. To this purpose, the proposed model uses a SPARQL Inferencing Notation (SPIN[1]) rule, shown in Listing 1, to detect this simple activity. As a first step, the rule checks if the *trackpoi:inArea* relation holds (line 8). This property relates a track ?*this* (i.e. *trackpoi:Track*), performing the activity, and a POI ?*poi* identifying a place. In this example, ?*this* should be a car and ?*poi* a road, in fact, the rule checks if ?*this* is a *trackpoi:Vehicle* instance (line 5) and if ?*poi* is a *trackpoi:Route* (line 7). The rule also checks if ?*this* speed (line 9) is greater than 0 (line 10), which means that vehicle is moving. In other words, if ?*this* instance is a track and ?*poi* instance is a route and the track ?*this* is moving (speed greater than 0), the *CONSTRUCT* clause holds, viz., the statement asserting that the vehicle ?*this* is running on route ?*poi* can be deduced.

[1] https://www.topquadrant.com/technology/sparql-rules-spin/

```
CONSTRUCT {
    ?this trackpoi:running ?poi .
}
WHERE {
    ?this a trackpoi:Vehicle .
    ?this trackpoi:track_ID ?id .
    ?poi a trackpoi:Route .
    ?this trackpoi:inArea ?poi .
    ?this trackpoi:speed ?s .
    FILTER (?s > 0) .
}
```

Listing 1: Running cars: the SPIN rule detects the simple activity *running* as triples stating that cars are running on a road

6.2. Complex activity detection through activity composition

Once the simple activities are detected, the system merges data from simple activities (carried out by one or more tracks and/or POIs) to define a complex activity, through a high-level description. More specifically, the knowledge about a simple activity, performed by a track, is combined with knowledge related to other activities performed by the same track or other tracks over time. Activities are first combined by location (if they occur at the same location) or in adjacent areas. In addition, they are also linked by time because complex activities are often composed of simple and consecutive activities over time. The collected knowledge describes complex activities that are more detailed and dependent on the scenario domain, such as the *crossing* activity which identifies a proper people's action strictly related to the road environment. As stated, complex activities combine more activities carried out by tracks over time. The simple activities are related to a frame and its time instant in the video. The *Activity Composition* module (Figure 1) not only has to check a combination of occurred simple activities for each track of an object, but also evaluates the temporal relations among them. Since a simple activity is defined as an instant timed (binary) relation (Definition 1), the complex activity is a collection of these simple activities/binary relations that hold in a time interval T, more formally:

Definition 2. ***Complex activity.*** *Let $\hat{M} = \{\hat{o}_1, \hat{o}_2, ..\}$ and $F = \{y_1, y_2, ..., y_p\}$ be the mobile and fixed object sets respectively, the complex activity of a mobile object $\hat{o}_i \in \hat{M}$ in a time interval $T = [t_1, t_2, \ldots, t_n]$ consists of time-related single activities S_t (with $t \in T$) carried out by the mobile object $\hat{o}_i$ and some object z in the time interval T:*

$$< C_T, \hat{o}_i, z >_T \;\; = \;\; S_{t_1} \wedge S_{t_2} \wedge ... \wedge S_{t_n} \tag{6}$$

where $z = \begin{cases} \hat{o}_j, & \hat{o}_j \in \hat{M}, \quad with \;\; j \neq i \\ y_h, & y_h \in F, \quad with \;\; 1 \leq h \leq p \end{cases}$

.

An example of complex activity, which needs to be detected over time, is the people crossing. Generally, to state that a person is crossing the road, there is a need to know if the person is on the road and if he/she is going to the

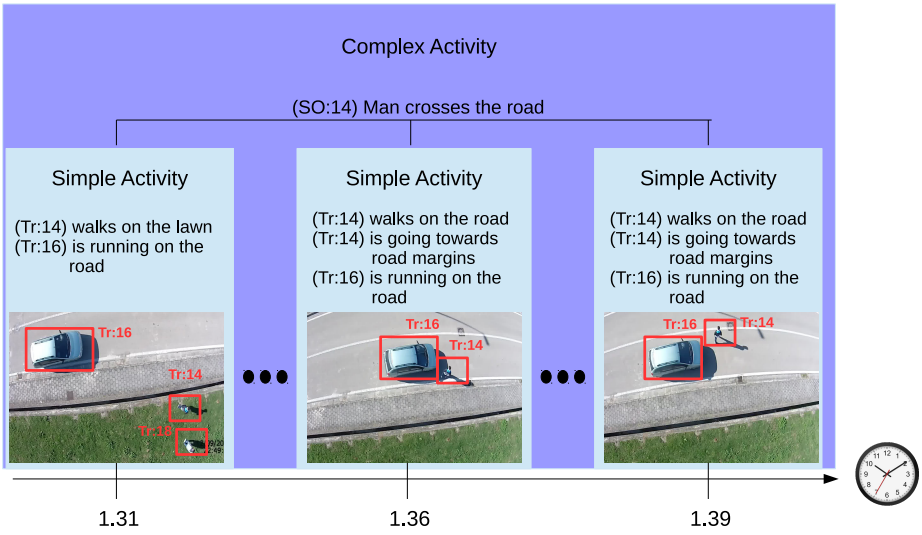

Figure 7: Activity composition: the complex activity *man crosses the road* is from three simple activities happening over the time $t \in [1.31, 1.39]$

other side of the road, otherwise the person is doing something else. In fact, people could keep staying on the road for many other reasons, for instance, for helping someone, i.e., police and rescuers if an accident is occurred, as well as, for working, i.e., road workers or reckless kids playing. The detection of a complex activity, such as crossing, requires the analysis of the mobile object evolution over time. Since the track represents the mobile object in a single video frame, the *Activity Composition* module collects all the tracks related to the same object and represents them as a unique instance of the *Scene Object* ontology class. In other words, the Scene Object (SO) individual represents the mobile object composed of a collection of all its representations (i.e., tracks) in the video frames. Since a track is associated with a specific frame/time instant of the video, track times associated to the SO individual provide entry and exit times of the object in the scene, as well as the time duration of the object stay in the scene.

Similarly, simple activities, directions and speeds associated with a track, are also collected for each SO individual through its tracks. Then, track simple activities are combined with respect to time and space, to identify the complex activity.

Figure 7 shows the activity composition for the *crossing* activity. Several simple activities are identified, each one detected for each track at a specific time instant of the video; in the example, the activities are detected in the time interval [1.31, 1.39]: for instance, at the time 1.31, the simple activities *walks on the lawn* and *is running on the road* are performed by tracks *Tr:14* and *Tr:16*, respectively. Once the simple activities are detected, tracks with the same identifier are collected to compose the SO individual, for example, the tracks identified by

Tr:14 over time (i.e. tracks *Tr:14* in 1.31, 1.36 and 1.39 instants), represent the SO individual *SO:14*. The different consecutive activities, carried out by the tracks compounding the SO individual, are collected; for example, the SO individual *SO:14* is composed of all the activities done by tracks *Tr:14*: *walks on the lawn*, at the time 1.31, and the *walks on the road* activities at the time 1.36 and 1.39. The time relations among the simple activities are fused with the direction (*SO:14* barely modifies its direction) and the spatial relations (*SO:14* moves to the opposite side of the road). The merging of the time-related simple activities (*walks on the lawn*, *walks on the road*) with the object features (direction, speed) and the contextual facts (moving to the opposite side of the road), supports the detection of the crossing people activity. Therefore, the system infers that *SO:14* crosses the road.

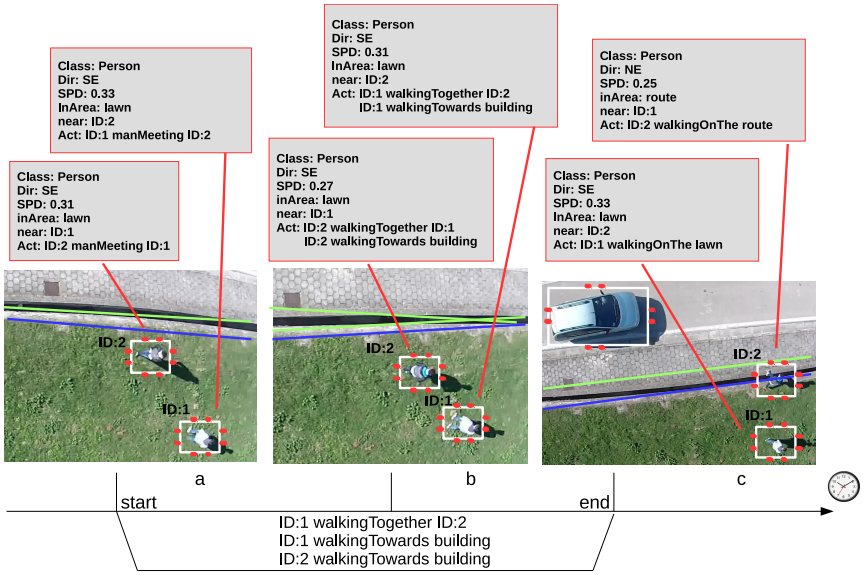

Figure 8: System at work: the *walkingTowards* and *walkingTogether* activities are detected. The functioning of the system is shown on a video scene showing two people walking together towards a place. The object annotations show the detected relations and activities.

7. A case study

In order to show how the system works, a case study is described. The video was shot on the road and it is part of our dataset [11], taken in our university campus. The focused scenario shows two persons meeting near a road, which decide to move together to some place in the surroundings; then, one of them, probably changing his mind, crosses the road on which a car is running. The video is given as an input to the system to detect the main activities carried out by the

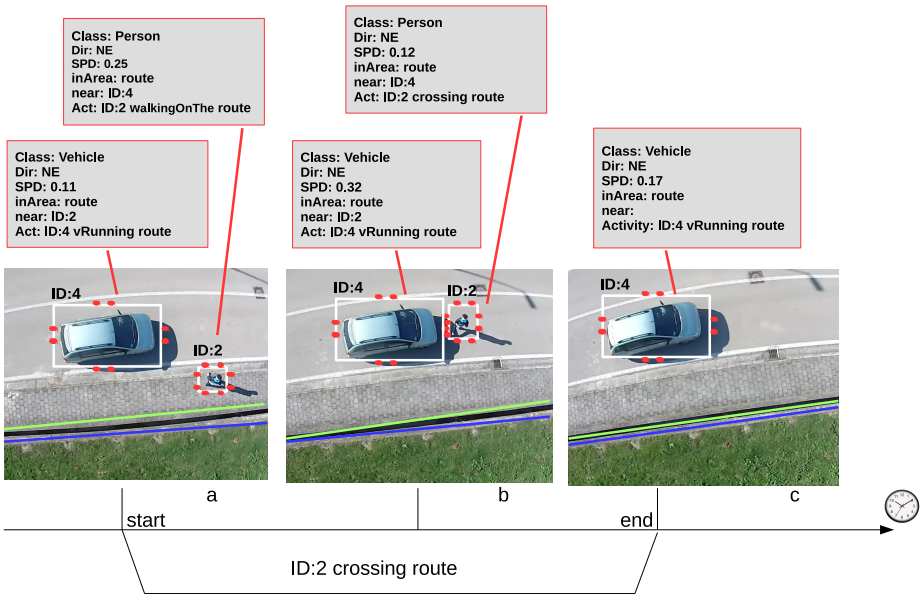

Figure 9: System at work: a crossing activity is detected. The functioning of the system is shown on a video scene showing a man crossing the road. The object annotations show the detected relations and activities.

people and vehicles. Figures 8 and 9 show the main output by our system on a processed video portion. As the first step, the system runs tracking and detects the moving objects in the video. Three people and several vehicles are detected throughout the lifespan video. Recalling the system overview shown in Figure 1, the modules *Object Classification* and *Area Classification* are involved in these activities. The *Object Classification* module labels the tracked objects, according to the object classification results. In the figure, tracks with identifier *ID:1* and *ID:2* are recognized as people, while the track *ID:4* is classified as a vehicle. Peculiar data about each track are calculated (i.e., speed, direction, width and height); the figures show the most significant ones. The *Area Classification* module performs area detection so that road and lawn areas in the scenario are correctly recognized. These areas are marked with graphical lines, and contours in the video.

Then, the tracking data along with the object and area classification data are provided to the semantic component (*Semantic Annotation and Reasoning*), which is in charge of the semantic annotation of the scenario with high-level information. It translates the data about tracked objects and POIs (i.e., in our case, the two people, the vehicle and the road shown in the scenario), into individuals of Track and POI classes, with the aim of populating the scene ontology TrackPOI.

Furthermore, this module semantically codes the spatio/temporal relations among the tracked objects, and between the objects and the POIs.

In the first part of the video (Figure 8), a *near* relation between the two people (i.e., track *ID:1* and *ID:2*) is found, then, later in the video, another *near* relation between one (of the two) people crossing the road and an oncoming car is discovered. Then, *inArea* relations between the people and the lawn, and then, between a person, a vehicle and the road are asserted as well. These relations are combined with track directions and speed to detect simple activities occurred in a frame. The merging of the speed and direction data of the two people with the *near* relation among them allows the detection of the *manMeeting* simple activity (Figure 8a). Similarly, the *inArea* relation between the vehicle and the road along with the vehicle movements detect the vehicle running on the road (*vRunning*) activity (Figure 9a).

At this time, the *Semantic Annotation and Reasoning* component composes discovered relations and simple activities (by reasoning on the knowledge base) to detect complex activities. Activity composition works the spatio/temporal relations with tracks, POIs and simple activities. Therefore, the *manMeeting* activity between tracks *ID:1* and *ID:2* is combined with people direction, speed, position over time and the environmental POI (i.e., the pub building on their direction). The system infers that the two people are moving together to the POI (*walkingTogether* and *walkingTowards* activities, see Figure 8b). These complex activities start when the *manMeeting* relation is found (Figure 8a), then, the people moving in the same direction, and almost the same speed allow the complex activity detection. These activities end when the *manMeeting* activity is no longer detected, and the directions and speeds change (see Figure8c).

Later in the video, *Activity composition* module combines the movements of the track *ID:2* (speed, direction) over time, with the spatio/temporal relations (i.e., *near*, *inArea*) and single activities (i.e., *walkingOnThe*) that hold between the track and the road, to infer that the person is crossing (Figure 9b). This activity composition is triggered by the detection of *walkingOnThe* simple activity (*ID:2 walkingOnThe route*), as shown in Figure 9a. The *crossing* activity for *ID:2* object lasts until *walkingOnThe* activity with the *route* is detected and no significant change in the direction is detected: in Figure 9c indeed, the *walkingOnThe* activity is no longer detected when the *ID:2* object runs out of the route and the scene. Let us notice that combining all the activities associated with a mobile object provides a complete scenario description: in the example, the person activity (*crossing*), the *near* relation between vehicle and person, the vehicle activity (*vRunning*), and its own features (i.e., speed and direction) allow the detection of a typical crossing scenario, without apparent risks (even though, the car and the person are very close to each other).

8. Framework evaluation

Two main evaluation tasks have been performed on the system, each one related to a macro-component shown in Figure 1: one experimentation is indeed related to the object and area recognition, the other to the activity detection. The system has been assessed on a dataset of annotated drone videos. The annotation comprises the presence of the events happening in the video, including

time, places, and IDs of the involved objects. The resulting accuracy for both the experimentations shows good results, evidencing that the synergy between low-level tracking methods and high-level semantic scene description leads to performance improvement of the overall system.

8.1. Dataset Description

The datasets employed for tests are composed by both videos recorded in our campus and downloaded from the Web[2]. They show scenes from several distinct outdoor environments such as roads, heliports, parks, etc. Also the UAV123 dataset[3] has been used in our experiments. Videos on our campus have been taken by using a DJI F-450 drone equipped with a Nilox F60 HD resolution camera. Tests have been carried out on 21 videos from these datasets and are selected, based on a similar length; they show different types of activities carried out by people and vehicles in different environments. Table 1 describes schematically all the information taken into account in our experimentation. Videos are grouped by the contextual environment appearing in the video (i.e. route, highway, parking lot, etc.), then, simple and complex activities detected from videos are listed in the corresponding columns (Table 1) along with a cumulative number of occurrences, given in the parenthesis. Detailed descriptions about the activities are reported in the Table 2 and Table 3.

8.2. Scene/Object video analysis module

The *Scene/Object Video Analysis* component has been tested so as to evaluate how accurate it is. In more details, the experimentation has focused on the object tracking, object classification and on the area categorization modules.

Detection and Tracking: The tests of the detection and tracking algorithms have been performed over the dataset acquired in our campus [11]. The experimental analysis has been performed from an effectiveness and an efficiency point of view. As for the effectiveness, the performance has been computed in terms of Precision (P), Recall (R) and F-Score (F):

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{9}$$

where TP, FP and FN represent, respectively, the number of True-Positives (TP), False-Positives (FP) and False-Negatives (FN). Such parameters have been computed by evaluating the overlapping between the boxes associated to the

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

Table 1: Dataset summary. Collected videos are arranged according to the enclosed context types (Environment). Simple and complex activities identified in each video are listed as well as the cumulative number of occurrences.

Environment	Video amount	Video average duration	Source	Simple Activities (# occurrences)	Complex Activities (# occurrences)
Route	4	1.55	Our videos, Web	vRunning(15), runningOff(4), overSpeedLimit(8), vehicleStopping(3), vehicleAccelerating(8), walkingOnThe(12), manRunning(3), walkingNear(8), walkingAround(8), manMeeting(5)	goingTowards(4), turnAround(2), avoidingObstacle(2), crossing(4), walkingTowards(2), walkingTogether(2), waitingFor(4) getsInTheCar()
Highway	3	1.32	Web	vRunning(8), runningOff(2), overSpeedLimit(5) vehicleAccelerating(8), walkingOnThe(2), walkingNear(1) walkingAround(3),	avoidingObstacle(2), walkingTowards(2), waitingFor(1)
Parking lot	3	1.27	Our videos, Web	vRunning(7), runningOff(6), vehicleStopping(4), vehicleAccelerating(5) walkingOnThe(7), manRunning(3) walkingNear(4), walkingAround(7), manMeeting(4)	goingTowards(3), turnAround(2), avoidingObstacle(3), walkingTowards(2), walkingTogether(4), waitingFor(3) getsInTheCar(3) getsOutOfTheCar(2)
Urban road	2	0.89	UAV123, Web	vRunning(14), runningOff(3), overSpeedLimit(2) vehicleStopping(7) vehicleAccelerating(12), walkingOnThe(14), manRunning(6), walkingNear(5), walkingAround(4), manMeeting(3)	goingTowards(6), turnAround(1), avoidingObstacle(4), crossing(3), walkingTowards(2), walkingTogether(4) getsInTheCar(3) getOutOfTheCar(2)
Park	2	1.46	UAV123 Web	walkingOnThe(14), manRunning(12), walkingNear(11), walkingAround(8), manMeeting(7)	walkingTowards(5), walkingTogether(6), waitingFor(4)
Heliport	2	1.14	Our videos	walkingOnThe(5), manRunning(3), walkingNear(2), walkingAround(4), movingObjects(), manMeeting(3)	walkingTowards(4), walkingTogether(2), waitingFor(3)
Crossroad	3	1.32	Web	vRunning(18), runningOff(6), overSpeedLimit(3) vehicleStopping(12) vehicleAccelerating(8), walkingOnThe(6), manRunning(3), walkingNear(7), walkingAround(3), manMeeting(6)	goingTowards(4), turnAround(2), avoidingObstacle(6), crossing(7), walkingTowards(4), walkingTogether(4), waitingFor(3) getsInTheCar(1) getsOuOfTheCar(4)
Other	2	1.32	UAV123	vRunning(9), runningOff(4), vehicleStopping(7), vehicleAccelerating(9), walkingOnThe(6), manRunning(4), walkingNear(7), walkingAround(9),	goingTowards(3), turnAround(2), avoidingObstacle(4), crossing(3), walkingTowards(4), walkingTogether(3), waitingFor(2) getsInTheCar(2) getsOutOfTheCar(1)

object at the ith frame, O_i, and the ground truth GT_i according to the Pascal Criterion:

$$\frac{area(O_i \cap GT_i)}{area(O_i \cup GT_i)} \geq 0.5 \tag{10}$$

otherwise O_i can be considered a FP. The following performance has been achieved: $P = 0.27$, $R = 0.18$ and $F = 0.21$.

As for the efficiency, we evaluate the processing time over a system on a module, namely an Intel Joule 570X board equipped with a quad core Intel Atom T5700 CPU@1,7 GHz (max 2,4 GHz) and 4 GB RAM LPDDR4. This platform has been chosen so as to have the possibility to mount the module directly on board of the drone, thus allowing the system to work in real time and to transfer only data instead than the whole video stream. With a 4CIF resolution (640×360), the algorithm is able to run at 16 fps, thus confirming the possibility to work in real time on this kind of board.

Objects classification: The samples used for our experimentation and collected from the datasets described above are about 4,050. Such samples have been manually labeled by a human operator and partitioned in three classes, namely 1553 samples of pedestrians, 1706 of vehicles and the remaining belonging to unknown objects. Some examples, for both pedestrians and vehicles, are reported in Figure 10. As we can see by analyzing the images, the samples have been acquired in different positions of the image, thus with different orientations with respect to the camera. Each class has been partitioned in training set (70%) and test set (30%), by paying attention to the fact that the same object (even if acquired in different position of the scene and then with different orientations) is not present in both training and test set. The accuracy achieved by the proposed system on the test set is 89.2%, thus confirming the effectiveness of the proposed approach.

Figure 10: Examples of persons (a,b,c) and vehicles (d,e,f) considered in our experimentation.

Area categorization: Starting from thee images contained in the dataset, we collected a dataset of about 30,000 patches, uniformly partitioned in grass, road and unknown samples. As for objects classification, each class has been divided in training set (70%) and test set (30%). The overall accuracy achieved on the test set is 95.3%.

8.3. Activity recognition

The object activities in the videos are mainly carried out by people and vehicles in different environments such as highways, urban roads, parking lot, parks etc., that can appear also in the same video. The most difficult activities to detect are those occurring in road scenarios, where the interaction between people and vehicles complicates activity detection. Our system performance has been assessed in recognition of simple and complex activities. Tables 2 and 3 show, respectively, the set of simple and complex activities considered in this experimentation.

Our experimentation is based on a ground truth of the identified activities, and some specific metrics have been designed to evaluate the accuracy returned by our system. The ground truth for a video lists all the occurred activities in chronological order of appearance in the video. Each activity entry provides information on the activity type (listed in Table 2 and Table 3), the scene object performing the activity and the place where the activity has been carried out. The starting and ending time of the activity are also included in the activity entry.

Our system detects the activities as triples written in the Web Ontology Language (OWL) [4]. Each triple consists of subject, property and object: the property name indicates the type of the detected activity, the triple subject says who (people or vehicle) performed the activity while the triple object represents where it happened.

Figure 11 shows a succession of activities, namely the system-detected (S) and ground truth (GT) activities, placed on the video timeline. Precisely, Figure 11a displays two activities, namely, *vRunning* (VR) and *manRunning* (MR), detected by our system and present in the ground truth. They are represented as boxes placed on video timeline: the box length describes the duration of the activity, and the time overlap among S and GT activities occurs when they are in front of each other, on the same portion of the timeline. Depending on the attained time match between the detected and ground truth activities, four possible comparison cases can be distinguished: (1) S and GT activities of the same type overlap temporally (for example, activities VR_1 and VR_a in Figure 11a), (2) S activity has no temporal overlap with any GT activities (i.e., MR_1), (3) S and GT activities overlap temporally but they are of different type (i.e. VR_2 and MR_a) and (4) GT activity does not find any temporal overlap with any S activities (i.e. VR_b).

These cases reflect the outcomes in terms of true positives (TPs), false positives

[4] https://www.w3.org/OWL/

Table 2: Tested simple activities.

Activity	Performer	Description
vRunning	Vehicle	Vehicle running on a place, generally a road
runningOff		Vehicle running off a place, generally the road
overSpeedLimit		Vehicle breaking the speed limit
vehicleStopping		Vehicle stopping
vehicleAccelerating		Vehicle accelerating
walkingOnThe	Person	Man walking in/on a place (i.e. road, park, heliport, square)
manRunning		Man running in a place
walkingNear		Man walking close to a place area (i.e. road, park, heliport, square)
walkingAround		Man walking around a place area (i.e. road, park, heliport, square)
movingObjects		Man pushing or carrying not living beings
manMeeting		Men meeting

Table 3: Tested complex activities.

Activity	Performer	Description	composed of (simple activities)
goingTowards	Vehicle	Vehicles going towards each other	vRunning, vehicleAccelerating
parking		Man parking vehicle in a parking lot or by roadside	vRunning, vehicleStopping, runningOff
turnAround		Vehicle turning around	vehicleStopping, vehicleAccelerating
avoidingObstacle		Vehicle avoiding another object	vehicleStopping, vehicleAccelerating
crossing	Person	Men crossing the road	walkingOnThe, manRunning, walkingNear
walkingTowards		Man going towards a place or POI	walkingOnThe, walkingNear, walkingAround
walkingTogether		People walking together	manMeeting, manRunning
waitingFor		Man standing in a area until a certain moment	walkingOnThe, walkingAround
getsInTheCar		Man gets in the car	walkingAround, walkingNear
getsOutOfTheCar		Man gets out of the car	walkingAround, walkingNear

(FPs) and false negatives (FNs) in precision and recall computation. As Figure 11b shows, true positives indeed are essentially the number of successful matches between temporally-overlapping S and GT activities of the same type (i.e. VR_a and VR_1). Let us remark that activities of the same type are carried out by the same scene object and happening in the same place. If a detected activity S does not temporally overlap with any other GT activities or, just it overlaps with GT activities of different types, it is considered as a false positive (see MR_1 and VR_2 in the figure). Similarly, a GT activity is considered as a false negative if it does not temporally overlap with any S activity or overlaps with S activities of different types (see VR_b and MR_a in the figure).
In case of detected activity, S and a ground truth activity GT of the same type have a temporal overlap (see Figure 11b, activities named VR_1 and VR_a, respectively), then S represents a true positive.

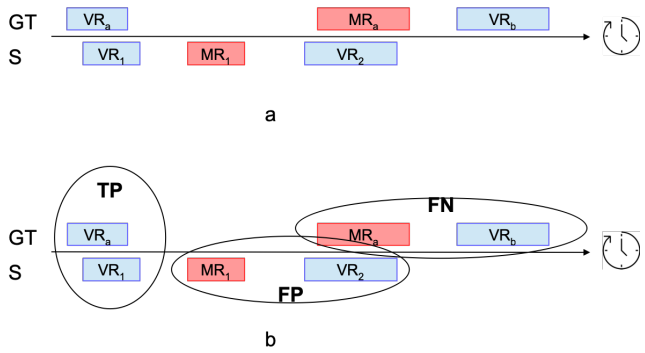

Figure 11: An example of activity comparison on vRunnnig (VR) and manRunning (MR): (a) temporal relations between the ground truth and detected activities, (b) True positive, false positive and false negative definition

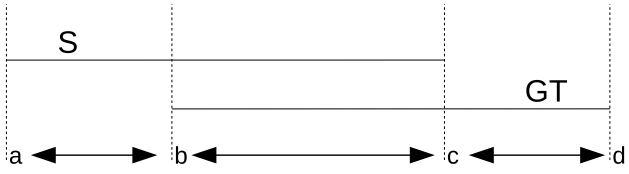

Figure 12: Temporal relations between a detected activity (S) and a temporally overlapping ground truth activity (GT) of the same type

The accuracy of our system is evaluated by using two accuracy metrics, that take into account the discovered temporal relations between detected and ground truth activities (of the same type). They have been used to evaluate the precision and recall of semantic activity recognition; they are described as follows.
Jaccard metric (JC) [26]: it is based on the comprehensive duration of the activity time and the overlapping time between a detected activity S and a

ground truth activity GT. According to Figure 12, JC calculates the ratio in seconds between the overlapping time among the two activities $(c - b)$ and the overall time covered by the two activities $(d - a)$, defined as follows:

$$JC(S, GT) = \frac{c - b}{d - a} \tag{11}$$

JC value for a detected activity S is compared to a prefixed threshold μ: if JC value is greater than or equal to μ, the activity S is assumed to be correctly detected and then considered as a TP. Otherwise, S is an FP; more formally:

$$TP_S = \begin{cases} 1, & \text{if } JC(S, GT) \geq \mu \\ 0, & \text{otherwise} \end{cases}$$

$$FP_S = \begin{cases} 1, & \text{if } JC(S, GT) < \mu \\ 0, & \text{otherwise} \end{cases}$$

The value μ is set to 0.2, accordingly to literature [26, 27]. In a nutshell, a JC-based TP is the number of the detected activities with JC value greater than or equal to μ. In case the detected activity S has JC value lower than μ, it is counted as an FP, and the relative activity GT is taken as an FN.

Mean Absolute Error Boundary (MAEB) metric [28]: it provides a value in the range $[0, 1]$ which represents how much the system-detected activity (S) overlaps with the ground truth activity (GT) of the same type. This value represents how much S can be considered as a TP. MAEB is different from the JC metric, that uses a threshold to select or not an activity as a TP; the MAEB value, indeed, is directly calculated according to the durations and the temporal overlap between the detected activity (S) and the ground truth activity (GT) of the same type, which are performed by an object in a place.

Figure 12 shows three different values which directly represent the extent to which the detected activity S is considered a TP or an FP, and the extent to which GT is considered an FN, more formally:

$$TP_S = \frac{c - b}{c - a} \tag{12}$$

$$FP_S = 1 - TP_S \tag{13}$$

$$FN_{GT} = 1 - \frac{c - b}{d - b} \tag{14}$$

Adding up all the TP_S values so calculated, for each detected activity S which overlap with the ground truth activities GT of the same type, the final TPs is calculated. In the same way, the total FPs and FNs are calculated as well.

As stated, TPs, FPs and FNs, determined with the two metrics, are employed to calculate precision and recall. Table 4 shows the precision and

Table 4: Test results on the detected simple and complex activities. Precision and recall, calculated with MAEB and JC metrics, are reported.

Activity	MAEB		JC	
	Precision	Recall	Precision	Recall
vRunning	0.94	0.86	0.95	0.91
runningOff	0.74	0.87	0.81	0.90
overSpeedLimit	0.94	0.87	0.97	0.91
vehicleStopping	0,71	0.89	0.76	0.94
vehicleAccelerating	0,78	0.90	0.84	0.94
walkingOnThe	0.87	0.93	0.92	0.96
manRunning	0.83	0.74	0.93	0.79
walkingNear	0.88	0.85	0.93	0.91
walkingAround	0.97	0.86	0.99	0.88
movingObjects	0.84	0.75	0.84	0.79
manMeeting	0.80	0.86	0.89	0.88
goingTowards	0.93	0.88	0.97	0.94
parking	0.86	0.92	0.94	0.92
turnAround	0.80	0.87	0.84	0.90
avoidingObstacle	0.77	0.84	0.81	0.86
crossing	0.80	0.86	0.84	0.86
walkingTowards	0.86	0.78	0.88	0.83
walkingTogether	0.78	0.75	0.83	0.77
waitingFor	0.85	0.81	0.88	0.84
getsInTheCar	0.92	0.88	0.94	0.92
getsOutOfTheCar	0.88	0.82	0.88	0.84

recall calculated with the MAEB and JC metrics on the simple and complex activities occurred in the video set. At first glance, results from the JC metric assume slightly greater values than those calculated with the MAEB metric, even though the performance is generally good for both the metrics. Let us notice that the precision in some cases is very high (i.e., greater than 90%): these values are obtained for several detected activities, such as *vRunning*, *overSpeedLimit*, *goingTowards*, *getsInTheCar* etc. High recall values greater than 90% are also obtained for activities such as *walkingOnThe*, *vehicleAccelarating*, *goingTowards*, *parking*.

The precision values obtained with JC are somewhat higher than the precision values obtained with MAEB; they are in correspondence with the activities *runningOff*, *manRunning* and *manMeeting*. In many cases, the two metrics, JC and MAEB, provide similar values for both precision and recall, or even identical (i.e. *movingObjects*, *getsOutOfTheCar*). MAEB-based results have almost the same precision and recall on simple activities (precision: 0.85, recall: 0.85) and complex activities (precision: 0,84, recall: 0,85), whereas the JC-based results have slightly greater recall on simple activities (precision: 0.89, recall: 0.89) than complex activities (precision: 0.88 recall: 0.86). By comparing the two metrics, on average, the MAEB-based results have a slightly lower precision (0.85) than JC-based results (0.89), while recall values are around 0.88 for both of them. MAEB metric is more sensitive to the variation of the durations and overlapping times of the detected and ground truth activities. Therefore, the MAEB-based results assume values very similar to the JC-based results, confirming that our system offers good performances, not only at recognizing the simple and complex activities but also at identifying their correct duration and occurrence in the video.

Our system reveals satisfying video content analysis, although the performance analysis in terms of real-time capability requires a further investigation. On short videos (one minute long and with a frame-rate equals to 25), real-time system performance looks promising for semantic annotation tasks. However, since the framework encodes information at the frame level, the system performance on longer videos is affected by the accumulation of data, whose semantic content is often redundant between successive frames. Our forthcoming task is indeed, to discard irrelevant knowledge at runtime (during the frame-by-frame generation of RDF triples) to speed up the complex activity composition, and hence, to enhance system performance and real-time replies.

9. Conclusion

The paper presented a hybrid UAV-based system that combines two research areas, Computer Vision and Semantic Technologies, to provide a high-level video understanding. The system is able to detect moving and fixed objects, to acquire the spatio-temporal relation among them and with the environment and, finally, to reconstruct the complete scenario from the activity viewpoint. The system is composed of two main components: the first one accomplishes Video Analysis tasks, it aims at detecting scene objects and the places where the objects move

by using classification methodologies. The other component employs Semantic Web technologies to encode video tracking and classification data into ontological statements: the built knowledge allows the generation of a high-level description of the scenario through activity detection. The main contribution of this paper focuses on modeling object activities at different levels of abstraction, which are then integrated to better describe the whole scenario. Simple activities are detected with respect to time, space and context. Then, they are composed together to obtain complex activities that allow a human-like characterization of the whole scenario. System components have been tested on several videos; the results are promising and confirm the potentiality of the approach.

References

[1] IHS, Top Video Surveillance Trends for 2016, https://technology.ihs.com/571965/top-video-surveillance-trends-for-2016, online; accessed 9 September 2018 (2016).

[2] Goldman Sachs, Drones. Reporting for works, https://www.goldmansachs.com/insights/technology-driving-innovation/drones/, online; accessed 9 September 2018 (2018).

[3] B. Nemade, Automatic traffic surveillance using video tracking, Procedia Computer Science 79 (2016) 402 – 409, proceedings of International Conference on Communication, Computing and Virtualization (ICCCV) 2016. doi:https://doi.org/10.1016/j.procs.2016.03.052.
URL http://www.sciencedirect.com/science/article/pii/S1877050916001836

[4] N. C. Mithun, T. Howlader, S. M. M. Rahman, Video-based tracking of vehicles using multiple time-spatial images, Expert Systems with Applications 62 (2016) 17 – 31. doi:https://doi.org/10.1016/j.eswa.2016.06.020.
URL http://www.sciencedirect.com/science/article/pii/S0957417416302998

[5] V. Bruni, D. Vitulano, An improvement of kernel-based object tracking based on human perception, IEEE Transactions on Systems, Man, and Cybernetics: Systems 44 (11) (2014) 1474–1485. doi:10.1109/TSMC.2014.2331217.

[6] V. Carletti, A. Greco, A. Saggese, M. Vento, Multi-object tracking by flying cameras based on a forward-backward interaction, IEEE Access 6 (2018) 43905–43919. doi:10.1109/ACCESS.2018.2864672.

[7] M. C. Chuang, J. N. Hwang, J. H. Ye, S. C. Huang, K. Williams, Underwater fish tracking for moving cameras based on deformable multiple kernels, IEEE Transactions on Systems, Man, and Cybernetics: Systems PP (99) (2016) 1–11. doi:10.1109/TSMC.2016.2523943.

[8] W. Min, Y. Zhang, J. Li, S. Xu, Recognition of pedestrian activity based on dropped-object detection, Signal Processing 144 (2018) 238 – 252. doi:https://doi.org/10.1016/j.sigpro.2017.09.024.
URL http://www.sciencedirect.com/science/article/pii/S0165168417303468

[9] Y. Li, Y. Guo, Y. Kao, R. He, Image piece learning for weakly supervised semantic segmentation, IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (4) (2017) 648–659. doi:10.1109/TSMC.2016.2623683.

[10] L. Zhao, Y. Zhou, H. Lu, H. Fujita, Parallel computing method of deep belief networks and its application to traffic flow prediction, Knowledge-Based Systems 163 (2019) 972 – 987. doi:https://doi.org/10.1016/j.knosys.2018.10.025.
URL http://www.sciencedirect.com/science/article/pii/S0950705118305112

[11] D. Cavaliere, V. Loia, A. Saggese, S. Senatore, M. Vento, Semantically enhanced uavs to increase the aerial scene understanding, IEEE Transactions on Systems, Man, and Cybernetics: Systems PP (99) (2017) 1–13. doi:10.1109/TSMC.2017.2757462.

[12] G. D'Aniello, M. Gaeta, T. P. Hong, Effective quality-aware sensor data management, IEEE Transactions on Emerging Topics in Computational Intelligence 2 (1) (2018) 65–77. doi:10.1109/TETCI.2017.2782800.

[13] L. Snidaro, J. García, J. Llinas, Context-based information fusion: A survey and discussion, Information Fusion 25 (2015) 16 – 31. doi:https://doi.org/10.1016/j.inffus.2015.01.002.

[14] J. Gómez-Romero, M. A. Patricio, J. García, J. M. Molina, Ontology-based context representation and reasoning for object tracking and scene interpretation in video, Expert Systems with Applications 38 (6) (2011) 7494 – 7510. doi:https://doi.org/10.1016/j.eswa.2010.12.118.
URL http://www.sciencedirect.com/science/article/pii/S0957417410014818

[15] G. Meditskos, I. Kompatsiaris, iknow: Ontology-driven situational awareness for the recognition of activities of daily living, Pervasive and Mobile Computing 40 (2017) 17 – 41. doi:https://doi.org/10.1016/j.pmcj.2017.05.003.
URL http://www.sciencedirect.com/science/article/pii/S157411921630195X

[16] J. Bernad, C. Bobed, E. Mena, S. Ilarri, A formalization for semantic location granules, International Journal of Geographical Information Science 27 (6) (2013) 1090–1108. arXiv:https://doi.org/10.1080/13658816.2012.739691, doi:10.1080/13658816.2012.739691.

[17] G. Okeyo, L. Chen, H. Wang, Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes, Future Generation Computer Systems 39 (2014) 29 – 43, special Issue on Ubiquitous Computing and Future Communication Systems. doi:https://doi.org/10.1016/j.future.2014.02.014.
URL http://www.sciencedirect.com/science/article/pii/S0167739X14000399

[18] F. Zhang, D. Zhang, Y. Liu, H. Lin, Representing place locales using scene elements, Computers, Environment and Urban Systemsdoi:https://doi.org/10.1016/j.compenvurbsys.2018.05.005.
URL http://www.sciencedirect.com/science/article/pii/S0198971517303903

[19] I.-H. Bae, An ontology-based approach to adl recognition in smart homes, Future Generation Computer Systems 33 (2014) 32 – 41, special Section on Applications of Intelligent Data and Knowledge Processing Technologies; Guest Editor: Dominik Ślęzak. doi:https://doi.org/10.1016/j.future.2013.04.004.
URL http://www.sciencedirect.com/science/article/pii/S0167739X13000642

[20] A. Salguero, M. Espinilla, Ontology-based feature generation to improve accuracy of activity recognition in smart environments, Computers & Electrical Engineering 68 (2018) 1 – 13. doi:https://doi.org/10.1016/j.compeleceng.2018.03.048.
URL http://www.sciencedirect.com/science/article/pii/S0045790617315483

[21] H. Y. Wang, Y. C. Chang, Y. Y. Hsieh, H. T. Chen, J. H. Chuang, Deep learning-based human activity analysis for aerial images, in: 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2017, pp. 713–718. doi:10.1109/ISPACS.2017.8266569.

[22] M. I. Ali, N. Ono, M. Kaysar, Z. U. Shamszaman, T.-L. Pham, F. Gao, K. Griffin, A. Mileo, Real-time data analytics and event detection for iot-enabled communication systems, Web Semantics: Science, Services and Agents on the World Wide Web 42 (2017) 19 – 37. doi:https://doi.org/10.1016/j.websem.2016.07.001.
URL http://www.sciencedirect.com/science/article/pii/S1570826816300324

[23] N. Li, H. Guo, D. Xu, X. Wu, Multi-scale analysis of contextual information within spatio-temporal video volumes for anomaly detection, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 2363–2367. doi:10.1109/ICIP.2014.7025479.

[24] D. Cavaliere, L. Greco, P. Ritrovato, S. Senatore, A knowledge-based approach for video event detection using spatio-temporal sliding windows,

in: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017, pp. 1–6. doi:10.1109/AVSS.2017.8078545.

[25] R. D. Lascio, P. Foggia, G. Percannella, A. Saggese, M. Vento, A real time algorithm for people tracking using contextual reasoning, Computer Vision and Image Understanding 117 (8) (2013) 892 – 908. doi:https://doi.org/10.1016/j.cviu.2013.04.004.
URL http://www.sciencedirect.com/science/article/pii/S1077314213000908

[26] K. Avgerinakis, A. Briassouli, Y. Kompatsiaris, Activity detection using sequential statistical boundary detection (ssbd), Computer Vision and Image Understanding 144 (2016) 46 – 61, individual and Group Activities in Video Event Analysis. doi:https://doi.org/10.1016/j.cviu.2015.10.013.
URL http://www.sciencedirect.com/science/article/pii/S1077314215002337

[27] A. Gaidon, Z. Harchaoui, C. Schmid, Actom sequence models for efficient action detection, in: CVPR 2011, 2011, pp. 3201–3208. doi:10.1109/CVPR.2011.5995646.

[28] P. Palmes, H. K. Pung, T. Gu, W. Xue, S. Chen, Object relevance weight pattern mining for activity recognition and segmentation, Pervasive and Mobile Computing 6 (1) (2010) 43 – 57. doi:https://doi.org/10.1016/j.pmcj.2009.10.004.
URL http://www.sciencedirect.com/science/article/pii/S1574119209000996