Università degli Studi di Salerno

Dottorato di Ricerca in Informatica e Ingegneria dell'Informazione
Ciclo 32 – a.a 2018/2019

Tesi di Dottorato / Ph.D. Thesis

Context-aware knowledge extraction for UV scene understanding

Danilo CAVALIERE

Supervisor: **Prof. Sabrina SENATORE**

PhD Program Director: **Prof. Pasquale CHIACCHIO**

Dipartimento di Ingegneria dell'Informazione ed Elettrica
e Matematica Applicata
Dipartimento di Informatica

Contents

"*A computer would deserve to be called intelligent if it could deceive a human into believing that it was human.*"

Alan Turing

The majority of this thesis is based on certain parts of the following publications. As a coauthor, I was involved actively in the research, planning and writing of these papers.

- D. Cavaliere, S. Senatore, M. Vento and V. Loia, "Towards semantic context-aware drones for aerial scenes understanding," 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Colorado Springs, CO, 2016, pp. 115-121. doi: 10.1109/AVSS.2016.7738062

- Danilo Cavaliere, Sabrina Senatore, Vincenzo Loia, Context-aware profiling of concepts from a semantic topological space, Knowledge-Based Systems, Volume 130, 2017, Pages 102-115, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2017.05.008.

- D. Cavaliere, L. Greco, P. Ritrovato and S. Senatore, "A knowledge-based approach for video event detection using spatio-temporal sliding windows," 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, 2017, pp. 1-6.
doi: 10.1109/AVSS.2017.8078545

- D.Cavaliere, V. Loia, S. Senatore, Data-Information-Concept Continuum From a Text Mining Perspective. pp.1-15. In Reference Module in Life Sciences, January 2018

- Cavaliere D., Loia V., Senatore S. (2018) A UAV-Driven Surveillance System to Support Rescue Intervention. In: Cerulli R., Raiconi A., Voß S. (eds) Computational Logistics. ICCL 2018. Lecture Notes in Computer Science, vol 11184. Springer, Cham, https://doi.org/10.1007/978-3-030-00898-7_8

- D. Cavaliere, S. Senatore and V. Loia, "Proactive UAVs for Cognitive Contextual Awareness," in IEEE Systems Journal. doi: 10.1109/JSYST.2018.2817191

- D. Cavaliere, A. Saggese, S. Senatore, M. Vento and V. Loia, "Empowering UAV scene perception by semantic spatio-temporal features," 2018 IEEE International Conference on Environmental Engineering (EE), Milan, 2018, pp. 1-6. doi: 10.1109/EE1.2018.8385272

- D. Cavaliere, S. Senatore and V. Loia, "A multi-perspective aerial monitoring system for scenario detection," 2018 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS), Salerno, 2018, pp. 1-6. doi: 10.1109/EESMS.2018.8405820

- Cavaliere, Danilo and Sabrina Senatore. Towards an agent-driven scenario awareness in remote sensing environments. 2018 IEEE Symposium Series on Computational Intelligence (SSCI) (2018): 1982-1989.

- D. Cavaliere, V. Loia, A. Saggese, S. Senatore and M. Vento, "Semantically Enhanced UAVs to Increase the Aerial Scene Understanding," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 3, pp. 555-567, March 2019. doi: 10.1109/TSMC.2017.2757462

- Danilo Cavaliere, Vincenzo Loia, Alessia Saggese, Sabrina Senatore, Mario Vento, A human-like description of scene events for a proper UAV-based video content analysis, Knowledge-Based Systems, Volume 178, 2019, Pages 163-175, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2019.04.026.

- D. Cavaliere, J. A. Morente-Molinera, V. Loia, S. Senatore and E. Herrera-Viedma, "Collective scenario understanding in a multi-vehicle system by consensus decision making," in IEEE Transactions on Fuzzy Systems. doi: 10.1109/T-FUZZ.2019.2928787

Abstract

In the surveillance systems, Unmanned Vehicle (UV) scene interpretation is a non-trivial problem, because UVs need to possess human-like common-sense knowledge to correctly interpret events and situations occurring in the monitored environment. Mobile camera-related issues, such as motion blur, can further complicate scene interpretation, causing a lack of reference points that badly affects the interpretation of scene entities and situations. To this purpose, this thesis investigates the synergistic combination of video tracking with Semantic Web technologies to enhance UVs at the interpretation of dynamic scenarios.

The first part of the thesis provides a survey conducted on the methods employed to extract knowledge from the acquired structured and unstructured data. When dealing with unstructured data, there is the need to define and process contextual data to extract high-level concepts from text. To this purpose, an approach is introduced to mine concepts from texts by building layered contextual knowledge on document terms exploiting a geometrical structure, called Simplicial Complex. Then, the focus switches to the knowledge extraction from multimedia data, and more specifically, video data. To this purpose, an ontology-based approach is presented to represent the video scene as composed of mobile (i.e., people, vehicles) and fixed entities (i.e., environmental sites and features), along with the spatio/temporal relations among them. The use of the ontology reasoning can support alerting event detection.

In the second part, solutions to detect high-level activities and events have been investigated. In order to build high-level descriptions of the activities and events, there is the need to build knowledge on various aspects of the scene at various levels of detail (scene object, environment, specific events, overall situation), additionally, there is a lack of general and reusable models to build knowledge on the various levels. In literature there are lots of

approaches that fuse data to spot simple or contextual events, but there are no models to relate them in order to fully describe the scene. Therefore, a composing approach is proposed to recognize complex activities from simpler activities carried out by the mobile entities in the scene. Then, a comprehensive situation detection approach is introduced. It defines a multi-ontology design pattern to incrementally build various layers of knowledge and depict the whole scene observed by the UVs, from the single scene entities to the high-level activities and situations.

The last part of the thesis analyses the scenario interpretation through systems employing multiple UVs and smart devices. Multiple UVs need ways to combine the knowledge they acquired to better interpret what happened. To this purpose, the previously introduced ontology-based framework has been enhanced with a novel agent-based model to allow UVs to cooperatively build knowledge on the scene. The employment of multiple devices can indeed provide better views on the scene to monitor, however, UVs not naturally come to an agreed scene interpretation. To tackle this issue, a consensus-based Group Decision Making (GDM) approach is proposed to support teams of UVs to robustly interpret the monitored scenario and evaluate the reliability of their interpretations.

Chapter 1

Introduction

In recent years, Unmanned Vehicles (UVs) have been widely used both in military and civil fields due to their capabilities of performing tasks in an automatic or semi-automatic way. Their success depends on the fact that they can be used to accomplish tasks, that can be too risky or difficult to perform, without directly involving humans. UVs can be of different types, such as Unmanned Aerial Vehicles (UAVs), Unmanned Ground Vehicles (UGVs) and Unmanned Underwater Vehicles (UUVs). Therefore, they can be used in applications set in different environmental contexts. Common applications include crowd monitoring, target searching, agriculture management, film making, public structure maintenance and more. Camera-equipped UVs have also been used to monitor objects moving in the video scene, such as people, animals, vehicles, for surveillance purposes. Computer vision algorithms have been employed to track the movements of the scene objects. To perform these tasks, camera-equipped UVs need to become aware of what they are observing to accomplish a specific task, they have been assigned with. In other words, camera-equipped UVs need to "understand" what they observed through their sensors. Therefore, situation comprehension requires UVs enhanced with human-like cognitive capabilities to deduct events and situations from video data.

To this purpose, this dissertation discusses various methodologies to allow smart devices to extract knowledge from structured and unstructured data. Then, several solutions are proposed to allow UVs to become aware of situations. The debated frameworks enhance UVs as knowledge-based systems, which use novel cognitive models to abstract knowledge on the monitored area. The presented approaches allow the UV to incrementally build knowledge from the collected data to reach a human-like description of the scenario. This UV feature can support human operators to monitor and analyse various situations and take action. This dissertation also tackles problems related to reaching robust scene interpretation by using system composed of multiple UVs and sensors.

1.1 Context and problem statement

The employment of UVs to accomplish surveillance tasks requires UVs capable of collecting information and defining their own interpretation of what is happened. In order to achieve scenario interpretation, humans can percept some elementary information, such as the presence of people, and detect some events. Then, humans use the logic, the experience and their own common-sense reasoning to relate the events and explain what happened in a scenario. As humans, UVs can percept information from the environment through their sensors, but they do not naturally come to a scenario interpretation. In fact, UVs lack the common sense reasoning capabilities to interpret the sensed raw data and abstract knowledge from them. Therefore, UVs, patrolling an area, need to “reason” over the data acquired to become aware of the occurred situations. Then, UVs require to be enhanced with cognitive capabilities to understand and relate situational elements (i.e., people, events, etc.) to achieve a robust situation awareness to accomplish their tasks. Let us consider a practical example: a human, observing a road environment, can immediately recognize the moving people and vehicles. Then, if the vehicle does not stop when a person crosses the road, the human mind relates the events by exploiting its experience with the common-sense reasoning ending up classifying the situation as potentially dangerous. In order to let a UV understand this scenario, it needs Computer Vision methodologies to recognize the moving objects and recognize them as people and vehicles. Then, the UV needs to analyse and recognize their movements and actions (i.e., people crossing, vehicle accelerating), this implies UVs possessing highly cognitive capabilities to contextualize people and vehicle movements with the environment, and analyse their interaction over time. Furthermore, to recognize the scenario as alerting, UVs need to possess knowledge on events and situations, that can be alerting, and use it to analyse the recognized events. Therefore, they must "know" that the co-occurrence of people and vehicles actions (i.e., people crossing, vehicle accelerating), along with their proximity, can put people's life at risks.

To understand a scenario, UV situation awareness requires, as first step, the acquisition of the main actors of the scene. Video tracking algorithms can support this step, performing the detection of mobile scene objects from frame to frame. The UV dynamism causes a lack of reference points that makes situation comprehension difficult to achieve. In fact, a mobile camera can cause problems to the object detection and tracking from video, as well as to the general interpretation of the events. For this reason, contextual information is strongly required to support UV comprehension of the video scene [1, 2]. Consequently, knowledge representation models are required to integrate video data with contextual data [2]. A robust scene knowledge representation also requires to model the space and time to understand the evolution of the scene. Therefore, scenario interpretation involves the comprehension of scene object movements and interactions to recognize events and situations from the observed scene. To this purpose, UVs need knowledge abstraction models to detect higher-level events from video data.

When using multiple UVs, the scenario comprehension becomes more and more challenging. The use of multiple UVs can indeed take the scene from many different angles and provide more useful information to depict the observed scenario. However, multiple

UVs can generate different interpretations of the observed scene. Therefore, there is the need of methods to guide UVs to a common interpretation of the scenario. UVs, usually, report contrasting information on the scene due to their different perspectives and features (i.e., different UV type, different applications). Many approaches, present in literature [3, 4, 5, 6], perform data fusion to integrate data coming from different UVs. Data fusion can certainly bridge UV information, but it cannot evaluate how the final group scenario interpretation satisfies each UV interpretation. Therefore, there is the need of methods to lead UVs to reach an agreed team interpretation of the scenario. Furthermore, UVs need tools to also evaluate how much the final group scenario interpretation satisfies all the UV perspectives on the scene, and, accordingly, evaluate the reliability of the team scenario interpretation. Another open problem is to determine which UVs in a team have the strongest impact on the determination of the team scenario interpretation.

We can summarize issues related to the UV situation awareness acquisition into the following questions:

- How UVs can acquire and represent knowledge from a video scenario ?
- How UVs can abstract knowledge on the scene to detect events and situations ?
- Do multiple UVs reach an agreed scenario interpretation that can better serve surveillance applications ?

1.2 Objectives

In this dissertation, our objective is to make advances in the field of UAV-based video surveillance by proposing novel knowledge-based frameworks to allow camera-equipped UVs to become aware of situations by starting from video data. Indeed, the achieved contribution in this thesis is motivated by the following assumptions:

- The existing approaches propose knowledge-based models to understand scenario from fixed cameras. These models exploit pre-fixed knowledge about the environment and the application to fulfil. Therefore, the approaches are not general-purpose and reusable.

- The mobile UV camera can generate several issues, such as prohibitive shots and motion blur, that make object detection and scenario recognition difficult to accomplish.

- The existing approaches focus exclusively on knowledge models that detect event and activities by fusing information retrieved from sensors. They do not propose solutions to abstract further knowledge to improve scenario description.

- When dealing with systems of multiple UVs, the main trends in literature focus exclusively on data fusion to support scene comprehension, but little has been proposed to evaluate UV information and, accordingly, reach a collective scene interpretation that satisfies all UV perspectives.

Consequently, the main contributions of this dissertation are targeted at alleviating UV issues to allow them to become aware of the observed scenario to support surveillance and monitoring tasks. Therefore, the main ultimate goal of this dissertation is to explore solutions to allow UVs to provide human-like interpretations of the scene. To this purpose, the discussion is aimed at answering to the following research questions:

- How can knowledge be extracted from structured and unstructured data ? Then, how can the extracted knowledge be organized to be easily exploited and reused ?

- How can UVs extract and represent knowledge on an evolutionary scenario to accomplish their tasks ?

- How can UVs abstract knowledge from tracking data to detect higher-level events and situations ?

- Can the combination of Computer Vision techniques with knowledge-based technologies lead to robust interpretations of the UV-monitored scene ?
- Multiple UVs can indeed provide different perspectives on the environment enriching the information for a comprehensive scene description. However, the information, collected by the UV from the environment, can lead to contrasting scene interpretations. Therefore, how can multiple UVs be guided to reach an agreed interpretation of the scenario they observed ?

1.3 Contributions

This dissertation contributes to solve the issues discussed in the previous section, by presenting novel knowledge-based frameworks for UV scenario comprehension. The proposed frameworks allow UVs to provide a human-like situation description. In details, they introduce:

- a knowledge-layered schema, based on the geometrical structure, to extract concepts from unstructured data
- an ontology-based knowledge representation of the scene, observed by UVs, in terms of mobile and fixed scene elements along with their relations.
- a module to detect activities and events from UV video through reasoning on spatio/temporal relations among the detected scene objects, and between them and the environment.
- a knowledge scheme of well-known ontologies to incrementally build knowledge on scene objects and activities to detect situation assessment by using Situation Theory.

- an agent-based modeling of UVs to allow them to build a global mental landscape of the scene and achieve comprehension.
- a GDM-based model to lead UVs to reach an agreed interpretation of the observed scenario.
- an application based on the introduced framework to get human-like information from a video that can support human operators in monitoring the evolution of monitored areas.

1.4 Organization of this dissertation

This dissertation is organized as follows. Chapter 2 introduces the Situation Awareness (SA) concept for humans and devices, and delineate the main SA features to support decision in complex dynamic environments. The main challenges in the mobile video surveillance and monitoring of evolutionary environments are presented. Then, Semantic Web technologies are also discussed to build knowledge on a domain.

In Chapter 3, an analysis of methods for knowledge extraction from structured and unstructured data is conducted. The analysis presents an incremental schema to classify methods according to the level of knowledge they build. Then, the chapter focuses on concept mining methods to extract knowledge from natural language texts. Among these methods, a method based on Simplicial complex geometric structure is described to build layered conceptualizations. The last section introduces the reader to the knowledge extraction from multimedia data, including features and issues.

Chapter 4 tackles the problem of extracting knowledge from multimedia data, specifically focussing on videos taken by camera-equipped drones. The first part of the chapter discusses how to alleviate the main issues related to knowledge extraction from drone videos by bridging Computer Vision with semantics. Semantic Web technologies are delineated to represent and generate knowledge on a scene observed by flying drones. Then, the rest of the chapter presents the TrackPOI ontology model to represent knowledge on the scene by modeling the tracked scene objects and contextual knowledge on the monitored environment.

After TrackPOI has been introduced, the Chapter 5 proposes to go further into the use of structured knowledge models for UV-based video surveillance. The chapter explores knowledge-based models to allow UVs to interpret the scene they observed. Firstly, a knowledge schema is introduced to allow the analysis and building of UV SA. Then, a preliminary extension of TrackPOI ontology is presented to model the scene events. As the next step, the chapter discusses the detection of activities, carried out by the detected scene objects. Thus, an activity detection approach is introduced to infer complex activities by composing simpler ones. Finally, the chapter provides an ontology design pattern to represent and support knowledge building at various layers from the video raw data on the detected objects to the high-level interpretation of the whole scene.

In Chapter 6, the focus moves to the use of multiple UVs and sensors to monitor evolutionary outside environments with the aim of interpreting the scene. The chapter introduces the main features and issues about multi-UV systems, including uses and capabilities of devices of different types along with methodologies to integrate and combine the knowledge they acquired from the environment. Then, the chapter introduces an agent-based model to support UVs in the definition of a mental representation of the scene, that integrates their knowledge to better interpret the monitored scene. According to the acquired data and perspective, UVs in a team can have different interpretations of the same scene. Therefore, they need to find agreement on the description that would better depict what they observed. To this purpose, an approach is analysed, that applies consensus-based Group Decision Making (GDM) to lead UVs to reach a robust scene interpretation.

Chapter 2

State of the art

2.1 Introduction

As stated in the previous chapter, UV scenario interpretation is a complex process composed of several tasks, such as object detection, environment detection and complex event detection. To accomplish these tasks, there is the need of solutions to observe dynamic environments, collect information and extract knowledge to reach Situation Awareness. Therefore, scenario interpretation requires to use different methodologies. Since many of the above mentioned tasks (i.e., event detection, situation assessment, etc.) are cognitively complex, knowledge-based approaches can be used to accomplish them. Historically, knowledge-based systems have been used for sensor data fusion and support scene object detection. This dissertation, instead, explores knowledge-based system potentials to support knowledge abstraction on the scene, starting from raw video sensor data, with the aim of making UAVs capable of providing human-like scenario interpretations. Obviously, receiving quick high-level responses can help human operators to improve the monitoring of multiple areas or desert areas, as well as the surveillance of urban or risky environments.

The employment of multiple UVs for surveillance can add further issues to an automated scenario interpretation. These issues are not only related to the UV fleet cooperation and coordination, but also concern the knowledge acquisition and evaluation for task accomplishment. In fact, multiple UVs need to share and compare their information to achieve an agreed collective scenario interpretation.

The reminder of this chapter introduces the Situation Awareness for UVs, knowledge-based systems and consensus-based GDM processes. It also discusses existing knowledge-based approaches to allow UVs to represent knowledge, detect activities and find an agreed interpretation of the scenario.

2.2 Situation Awareness

According to Endsley [7], the Situational Awareness (SA) is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future. SA is recognized as a critical foundation for robust decision-making in many fields, such as air traffic control, military command, emergency response and more. A scarce SA has been demonstrated to be one of the primary factors of human error in accidents.

According to Endsley definition [7], SA can be defined into achieving three states corresponding to three distinct levels of knowledge:

- **Perception (level 1 SA)** concerns the perception of the status, attributes and dynamics of the relevant elements in the environment. Perception is basically a simple recognition of situational spatial elements.

- **Comprehension (level 2 SA)** refers to a higher level of knowledge, which is achieved by integrating level 1 SA elements, through pattern recognition, evaluation and interpretation, to understand how the generated information impacts on prefixed goals. This integration process is aimed at achieving a comprehensive picture of the world or part of it.

- **Projection (level 3 SA)** is the highest SA level, which consists in the ability to project the actions of the situational elements in the future. Once the perception of the elements (level 1 SA) and the comprehension of the situation are accomplished (level 2 SA), the projection of the situational elements in the future allows the evaluation of the possible impacts on the environment in the near future.

The SA states represent different levels of knowledge, SA systems refer to the level of SA achieved in a team and SA processes, sometimes referred as situational assessment, refer to processes to acquire knowledge and update the SA state. Endsley's SA model employs several variables that can impact on SA building and maintenance. These variables include the individual, who has to acquire SA, the environment and the specific task the individual must accomplish. For instance, different individuals can have different abilities to acquire SA. Therefore, different individuals can achieve different SA, even though they use the same SA system and training. For what concerns the main purpose of an SA system, Endsley points out that SA "provides the primary basis for subsequent decision making and performance in the operation of complex, dynamic systems" [7]. Therefore, the overall objective of an SA system is basically to build a thorough knowledge to support decision, covering cue recognition, situation assessment and prediction at basis of a good decision process, but it is not enough to make decision itself. Beyond the variations related to individuals, environment and tasks, an SA system strongly depends on time, especially when SA is required to make decision in time-critical scenarios. SA can vary with respect to changes in the environment, individual actions and task characteristics over time. Individuals, who have to perform a task, need to build and update their own mental representation according to time, updating their plans and actions as new input data are acquired. SA also involves spatial knowledge about the events and activities occurred in specific locations. Consequently, according to Endsley's model, SA requires the perception, comprehension and projection of situational information, as well as spatial and temporal knowledge.

When tasks are assigned to a team of individuals, team members need to build global awareness. Since tasks assigned to individuals can overlap, they need to share their knowledge (shared SA).

SA has not only been applied to humans, but also to non-human individuals, such as electronic devices (i.e., smart sensors, robots, etc.). Many trends in literature apply SA to mobile devices to help them to accomplish tasks, such as formation, mission control and navigation [8]. The application of SA to devices allows them to increase their knowledge on the environment, and, consequently, raise up their level of autonomy. According to SA definition, if the device goes from lower to higher SA levels, it can reach higher levels of autonomy due to a thorough knowledge acquired.

Among devices, let us consider sensor-equipped UAVs. The application of the SA definition to a UAV allows the definition of the UAV Situation Awareness into the three distinct SA levels:

- **UAV Perception** refers to UAV ability to sense mobile and fixed elements in an environment. Technically, the UAV percepts situational elements through sensor data acquisition.
- **UAV Comprehension** concerns the UAV capabilities to process the sensor data on situational elements with some methods, that allow the UAV to achieve higher-level comprehension of the whole environment.
- **UAV Projection** involves the UAV ability to project the current environment state in the future.

This framework allows the analysis and building of the UV Situation Awareness.

2.3 Semantic Web

The Semantic Web is an extension of the World Wide Web proposed by Tim Berners-Lee to allow computers to automatically reuse, retrieve and reason over a web of data. The main aim of Semantic Web is to extend web resources with meta data expressing

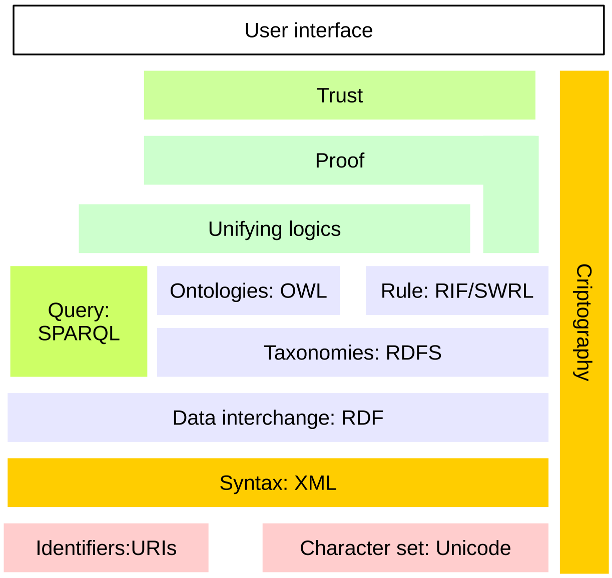


Figure 2.1: Semantic Web Stack

a meaning about them. This way, computers can process humans , according to Tim Berners-Lee [9]:

> I have a dream for the Web in which computers become capable of analyzing all the data on the Web, the content, links, and transactions between people and computers. A "Semantic Web", which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize.

According to this idea, computers can act as intelligent or semantic agents, capable of understanding the content of web resources and smartly exchange and reuse the acquired information. Therefore, the semantic agents need to understand the knowledge expressed in web documents to relate them. This way, the semantic agents can perform automatic searches and lead the human users to find the right information they were looking for. To allow this semantic enhancement of Web, the Semantic Web offers a set of data formats and languages to represent, query and infer knowledge from documents. Therefore, the Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [10]. The Semantic Web stack reports all the formats and technologies, as components of a unique framework, introduced to allow the collection, structuring and recovery of linked data. These technologies provide formal descriptions of concepts, terms and relationships among them within a given domain. The complete Semantic Web Stack is shown in Figure 2.1, the components are:

- *XML* provides a basic syntax for web documents
- *Resource Description Framework (RDF)* is the fundamental language to express data models, representing the web resources and their relationships.
- *RDFSchema (RDFS)* extends RDF with a vocabulary of principles to define classes and properties about the RDF-based resources.
- *OWL* provides an advanced vocabulary to model further features and relationships about classes and properties.
- *SPARQL* is a language for querying web data sources.
- *RIF and SWRL* are web rule languages to run rule-based inference over the web sources.

2.3.1 Ontologies

The proliferation of textual information makes the extraction and collection of relevant information a tricky task. Although search engines are recently enhanced with Artificial Intelligence techniques, the vagueness of natural language is still an open problem. At present, the ontology has proven itself to be an effective technology for representing as a form of concepts, the web information and then sharing common conceptualizations that are referenced as knowledge. An ontology gathers concepts from the real world by means of unambiguous and concise coding. At the same time, it should capture the terminological knowledge that sometimes embeds imprecise information, should support the management of semantic data and the intrinsic ambiguity in their model theoretic representation, provide enhanced data processing and reasoning, and then supply a suitable conceptualization that bridges the gap between flexible human understanding and hard machine-processing [104]. Simple ontology schemas, called taxonomies reflect vocabulary properties (such as term definitions, constraints and relationships) are often used as semantic models representing hierarchical classification of concepts. A taxonomy describes relations between related concepts as super-sub category or subsumption relationship. This schema enables to represent articulated concepts as subsets of more simple concepts, and create a layered structure based on concept complexity useful for concept analysis. In most cases, taxonomies are represented by hierarchical tree structure of classifications for a given set of objects.

In order to represent concepts, an ontology can be seen as a semantic network composed of nodes and edges to link nodes, where the nodes represent the concepts and the edges represent the relationships among concepts. Abstract concepts are represented by ontology classes, therefore, a real concrete example of an abstract concept is represented by class instance. The relations among two different concepts are represented by ontological properties. Ontological axioms are represented in the form of triples subject-predicate-object, in order to represent that a resource (subject) has a property (predicate) which assumes as value another resource or a literal (object).

The model-theoretic semantics behind the Semantic Web are based on Description Logics (DLs), in order to model assertions and perform reasoning on the ontological schema, in order to produce new axioms and increase the knowledge about the concepts.

Coding (meta-) data and relations between them into an ontology, starting from unstructured text is a necessary step towards the knowledge modeling [105]. Ontologies and ontology-based applications [106] achieve language processing for extracting keywords inherent to domain concepts from natural language documents. Semantic Web technologies yield knowledge in the form of concepts and relations among them; they translate the vagueness of natural language (embedded in the linguistic terms) by identifying conceptual entities in the resource content. Intelligent AI computing proactively supports these activities, modeling this ambiguity by more suitable methods and techniques that natively reflect the uncertainty and the reasoning of the human thought process [104].

The nature of the ontology modeling moves towards a shared conceptualization and a consequent reuse of the same data, reinforcing the request that data about concepts and their relationships must be specified explicitly and data needs a robust formalization.

Due to the stringent reliance of applications on well-designed data structure, semi-automatic tools like OntoLearn[1], AlchemyAPI[2], Karlsruhe Ontology (KAON) framework[3], Open Calais[4] and Semantria[5] are widely developed, to automatically extract entities, keywords and concepts from unstructured texts.

Systems and applications as KnowItAll [107], DBpedia [108], Freebase [109] also provide publicly semantically annotated knowledge resources; some others such as ConceptNet [110], Yago [111] aim at conceptually capturing common sense knowledge; sometimes, due to the quality bottleneck, projects like Cyc, OpenCyc [112] and WordNet are often built on manually compiled knowledge collection.

[1] http://ontolearn.org/

[2] https://www.ibm.com/watson/alchemy-api.html

[3] http://kaon2.semanticweb.org/

[4] http://www.opencalais.com/

[5] https://www.lexalytics.com/semantria

2.3.2 Fuzzy ontologies

Ontologies have been widely employed as a knowledge representation model for multi-agent systems and have played a crucial role in the development of the Semantic Web. Typical ontologies represent facts as axioms exploiting a two-valued semantics. The imprecise and vague nature of the real world applications needs an extension of the traditional logic behind ontology modeling [11], with a more flexible, fuzzy version of it [1]. Axioms, classes, instances and properties present in the traditional ontologies, are re-defined in fuzzy ontologies as fuzzy axioms (A), fuzzy concepts (C), instances (I), relations (R) and fuzzy relations or roles (F). The fuzzy ontology can be defined as a quintuple $Q_F = (I, C, R, F, A)$ [12]. Fuzzy ontologies use fuzzy modifiers and quantifiers [13]. In ordinary crisp ontologies, an instance can be of a specific type (class) or not. The fuzzy concept, instead, allows a soft class membership, expressed with a degree of truth. For instance, given the *Deep_ river* class instances: *Amazon* and *Danube*; and their class memberships: *(Danube a Deep_ river 0.81)* and *(Amazon a Deep_ river 0.45)*; it can be stated that the Amazon River is quite deep, but not as deep as the Danube.

FuzzyDL reasoner [13] provides a language to define and reason over fuzzy ontologies, by using variables which can assume a degree of truth. Axioms can be combined in a more complex way to build a fuzzy knowledge base. Fuzzy operators can be applied to the concepts, in particular the aggregation operators provide a powerful tool to model complex domains by aggregating simpler fuzzy concepts.

Inference on the fuzzy knowledge base allows to check knowledge base consistency, concept subsumption and concept satisfiability, as well as to support the variable optimization and the computation of defuzzification [13]. In our approach, Maximum Concept Satisfiability queries are used. Maximum Concept Satisfiability queries involve Concept Satisfiability [13]:

Concept Satisfiability. Let C be a fuzzy concept in the knowledge base K, and D a degree of truth, C is said to be D-

satisfiable, with respect to K, if there exists some instance with degree greater than or equal to D.

FuzzyDL extends the Concept Satisfiability definition to a specific instance o instead of an arbitrary one. Then, the Maximum Concept Satisfiability query infers the maximal degree to which the concept is satisfiable (best satisfiability degree):

Best satisfiability degree. Let C be a fuzzy concept in the knowledge base K, and D a degree of truth, the best satisfiability degree of C is the maximal degree D such that C is D-satisfiable.

The sapplication of the Maximum Satisfiability for the concept C over the specific instance o assesses how much o satisfies C.

Since experts' opinion in GDM problems are vague for their nature, fuzzy ontologies better deal with imprecise experts' judgements [14, 15]. Fuzzy ontologies make it also possible to store large amounts of data [14], represent and merge interpretations of different aspects [15]. Fuzzy ontologies are also employed in information retrieval [16], ambient intelligence [1], image interpretation [17], ontology merging [18], recommendation systems [19] and decision-making [15].

2.4 Multi-UV scenario acquisition

In multi-UV monitoring systems, UVs need to achieve a robust and unambiguous interpretation of what they observed. Methods to lead a group to an agreed outcome have been studied in Group Decision Making (GDM).

2.4.1 GDM with consensus modeling

A GDM problem may be defined as decision situations where there is a problem to be solved by two or more experts [20]. Therefore, a GDM problem can be described as follows. Let $E = \{e_1, \ldots, e_n\}$ be a set of experts and $X = \{x_1, \ldots, x_m\}$ a set of alternatives. A GDM problem is to sort X using the preference values P^k, $\forall k \in [1, n]$, provided by the experts. When having to provide preferences, there is an expert-system communication gap. The

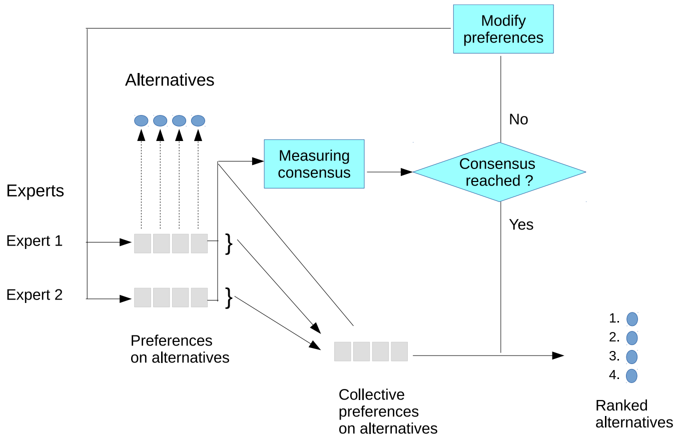


Figure 2.2: GDM process.

system prefers to manage numerical values while the experts are used to express himself/herself by using imprecise information such as “good”, “bad”, etc. Therefore, there is a need of methods that help systems to understand imprecise information provided by the experts.

In order to deal with vague and imprecise information, fuzzy set theory has been widely investigated in literature [21, 22], to represent the linguistic terms by means of linguistic variables, whose values are not numbers but words in natural language. Expert preferences can be expressed as terms from a fuzzy linguistic variable. In [21], the linguistic terms are used to model complex linguistic expressions, for qualitative decision-making. In [22], the consistency of different types of reciprocal preference relations have been studied, which are expert preferences constructed by comparing the alternatives. Expert preferences, expressed as fuzzy linguistic variables, [23], can be aggregated to build collective preferences by using well-known aggregation operators, such as the weighted means [24, 25], the Ordered Weighting Averaging (OWA) [26], the Linguistic Ordered Weighting Averaging (LOWA) [27], etc. Many works [24, 25] propose these operators to aggregate preferences; they present some generalized weighted averaging aggregation operators to aggregate intuitionistic fuzzy sets [24], or pythagorean aggregation operators for enhanced decision-making approaches [25].

Consensus Reaching Processes (CRPs) can be applied to the GDM problems to help experts reach an agreement [28]. Figure 2.2 shows the GDM process employing a CRP: the experts provide their preferences on the alternatives, which are aggregated to build a collective preference. The consensus is calculated and reached iteratively: if the experts reach consensus, the built collective preference is used to rank the alternatives (selection process). Otherwise, the process keeps running by asking the experts to modify their preferences to achieve a solution with a higher consensus. In the CRP process, an external moderator leads the experts to reach the highest consensus and keeps the most of the involved experts in agreement among them. In some approaches, the consensus achievement depends on a prefixed threshold value [28]. Many trends [29, 30, 31] in literature studied the application of CRPs to GDM problems. In [29], a series of criteria is defined to analyze and compare the efficiency of different CRPs. In [30] a novel CRP with individual consistency control is proposed to avoid repeating the time-consuming consistency improving process after the application of CRP. Weighing the expert opinions give them more importance: their opinion is interpreted as more reliable than others or more highly experienced to solve the problem [31].

Chapter 3

Knowledge extraction from structured and unstructured data

3.1 Introduction

The knowledge extraction is a complex activity of identifying valid and understandable patterns in data. When dealing with unstructured data, these patterns are often related to the Natural Language Processing tools: the text content is parsed to identify topics that could be described by single terms, enhanced terms matching by adding phrases, complex key-phrases. Extracting the relations between terms, or verbs and their arguments in a sentence has the potential for identifying the context of terms within a sentence. Contextual information can serve high-level concept extraction, as it will be discussed in the first part of this chapter (Section 3.2).

In texts, the fundamental first step to knowledge extraction is the interpretation of the natural language. When dealing with mobile devices, such as robots and drones, the output data is a multimedia file, such as an audio or video file. These file types are structured, therefore, they report the device output in a specific format rather than in natural language. Data in multimedia files can refer to specific law level data, which can be relative to sensors or other features of the device, or to higher-level information. Therefore, knowledge extraction from a multimedia file requires specific knowledge about the data and models to allow their integration and interpretation. Knowledge extraction from a multimedia files is particularly challenging because it can support many applications with smart devices, such as the employment of drones or robots in video surveillance. To this purpose, knowledge extraction from multimedia files has been investigated in aerial video surveillance applications and discussed in the second part of this chapter (Section 3.3). Before going into details about knowledge extraction from texts and multimedia files, the rest of this section analyses some knowledge extraction approaches according to a layered knowledge representation.

3.1.1 Knowledge extraction: from words to concepts

The contextual information can be very informative for capturing the actual meaning of some terms, and the sense relations involved in the case of polysemy. The information about "who is doing what to whom" reveals the role of each term in a phrase and the higher level meaning of the phrase. Simple terms or complex expressions represent different granularity levels of the knowledge that can vary depending on the formal methods used, the final conceptualizations and to the intended meaning behind the sentences, whose interpretation often escapes automated machine-oriented approaches. Figure 3.1 shows our representation of the knowledge continuum, in an incremental layer-based transformation. The knowledge schema is composed of more granularity levels, starting from "atomic" entities, i.e., single words, to reach a high level representation of knowledge as ensemble of conceptualization and semantic correlations.

The lowest layer represents the primitive knowledge related to single words and terms in a document. Word collection per document is considered the basic data, or more simply, the *data*. The

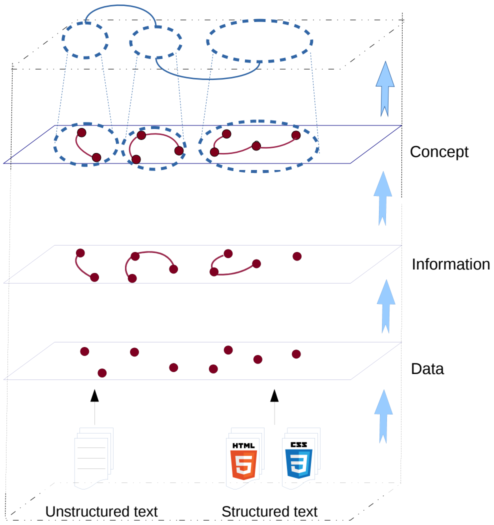


Figure 3.1: Layered representation of the knowledge: from words to concepts

Data layer consists of raw data, which are generally composed by single words extracted from textual documents. These documents can be unstructured, i.e., the content is plain text, or structured, such as web resources with markup annotations (with standard languages such as XML, HTML, CSS, etc.).

In other words the unstructured texts contain just words, while the structured ones contain meta-information like HTML tags, which can be used to extract more information about the concepts expressed by the text and the document structure. Let us consider the following document extract:

> *Canadian pop star Michael Buble married Argentine TV actress Luisana Lopilato in a civil ceremony on Thursday. The Grammy-winning singer of "Crazy Love" and his Argentine sweetheart posed for a mob of fans after tying the knot at a civil registry in downtown Buenos Aires.*

Approaches modeling the *Data* layer directly work on the single words such as: “star”, “actress”, “TV”, etc. and often only nouns (and adjectives in some cases) are precessed: since these approaches work on the term frequency, proper nouns can be discarded if no named entity task is foreseen in the process. Usually, flat term ensembles are produced by this layer.

When data are furthermore processed or structured, the informative granule increases. The *Information* layer consists of more data structuring which considers linguistic and grammar relationships among atomic terms. More articulated sequences of words, often called keyphrases are extracted by the text analysis. Stemming, lemmatization, part of speech tagging are the basic NLP tasks involved; they allow the identification of terms that are linked by relations: sequences of only nouns, combinations of adjectives and nouns, named entities, etc. characterize the *information.*

By considering the previous extract, named entities such as “Michael Buble’, “Buenos Aires”, and a keyphrase such as “pop star”, or syntactic relations between terms or entities, i.e., “civil ceremony” are the candidates to describe the information granule generated by this layer.

The highest layer of knowledge corresponds to a further and articulated structuring of the data which represent a more detailed and high-level knowledge. The informative granules of this layer define complex structure of terms that are supported by external sources, such as lexicons, knowledge bases and ontology-based schemas in order to provide richer conceptualization, specialized thanks to contextual information and the terms relationships. Compositions of simpler linguistic expressions yield complete and complex descriptions of concepts that, at this stage, are well-defined. The *Concept* Layer produces correlations of terms so rich that clearly identify a conceptualization, often specific in a given domain (according to the content of the processed document collection) and assumes a clear connotation for an authentic interpretation of the textual content. A crucial role is played by external knowledge-based sources, especially if they are semantically annotated, because, they yield additional (often inferred) information

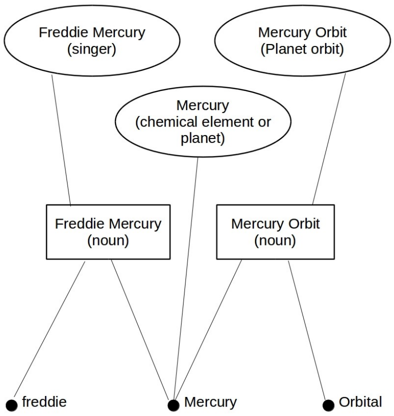


Figure 3.2: Knowledge stratification example

to extend, enrich and disambiguate concepts extracted by text. The extracted concepts, connected to each other by term-based relationship, compound a wider semantic network that represents the highest granulation layer.

The Concept layer can deduct more articulated concepts as facts from text; recalling the previous extract, the ex-novo concept $<< marriage >>$ coming from the named entities "Michael Buble" and "Luisana Lopilato" could be extracted exploiting an ontological schema, thus enriching the initial more vague concept.

Each knowledge level introduces further relations among more articulated and high level data are taken into account. These relations are useful to provide a better contextual insight. To this purpose, let us consider a further example of knowledge stratification, given in Figure 3.2, which considers the three words "mercury", "orbital" and "freddie". At the *Data* level, these words are considered, according to the proposed conceptualization, as simple three words or singleton terms $<< mercury >>$, $<< orbital >>$, $<< freddie >>$ (see words beside the bullets at bottom of Figure 3.2). Terms or words, at the *Information* level, are further structured according to term relationships and/or other transformations (e.g. POS tagging). Thus, terms appearing close to each other within a document can be recognized as named entities or linguistic period structure (e.g. noun, verb, etc.). In this case if the words “freddie” and “mercury” are very close and used as phrase subject or object, $<< freddie\ mercury >>$ will be recognised as proper noun (see texts in rectangles, Figure 3.2). Finally, these more structured data can be further structured at *Concept* level. At this level, the data include more contextual information about terms on different documents. The word “mercury” , at this level, can refer to different concepts: the chemical element, the singer if close to “freddie”, or the planet in the solar system if contextual related to “orbital” (see texts in the ellipses, Figure 3.2).

In a nutshell, a strongly connected structure may be figured from the layer-based knowledge model shown in Figure 3.1: all the informative granules are linked, through all the granulation levels, in order to generate a large knowledge base that comprehensively describes the entire text documents domain.

3.1.2 An analysis of Data-, Information- and Concept-driven approaches

In the light of the layered stratification of the knowledge, described in Section 3.1.1, the frameworks and tools, designed for knowledge extraction from documents, can be analyzed and classified according to the introduced layered knowledge model. To this purpose, some salient features/aspects have been selected to evidence peculiarity and/or similarity in the basic approaches, at each knowledge layer. Tables 3.1, 3.2 and 3.3 show indeed the main frameworks, whose methodologies and implementation design produce a knowledge model that better reflects a knowledge layer of Figure 3.1. The selected features are mainly five, described as follows.

- **Research sub-field**: this feature identifies the research areas where the framework and tools are located, according to the methodologies, functionalities and techniques employed. The feature highlights the synergies between the different areas involved in the knowledge structuring, through an incremental informative granulation that yields the knowledge representation. In tables, the first reported field is a macro-field which the approach deals with. Three macro-fields have been considered: Natural Language Processing (NLP), Information Retrieval (IR) or Semantic Web (SW).

- **Knowledge representation**: reports formal methods used for the knowledge extraction from text. It presents the methodologies and the tecnologies employed to represent and model the knowledge extracted.

- **Ontology-based support**: evidences the role of external ontologies or ontology-based tools, in supporting the concept-based knowledge representation. Referencing to concepts of existing ontologies to describe entities is becoming a common practice in Text Mining.

- **Similarity measure**: presents a primary feature, aimed at discovering low-level informative granules of knowledge. The similarity is the basic measure to compare text entities, such as words, terms, named entities, concepts, targeted at discovering syntactic or semantic relationship. Syntactic similarity concerns the sentence structure, exploiting for instance, the root or the lemma of a term; the semantic similarity is more complex to elicit: it discerns the correct sense (or concept) behind the term (or sentences) to get the contextual, actual meaning.

- **Semantic annotation support**: semantic annotation is a new way to represent knowledge in the form of concepts, which is far from the textual annotations on the content of documents. The feature indicates if the annotation is retrieved by pre-existent or ad-hoc defined ontologies. Tools that achieve IR tasks using semantic web technologies often carry out annotation tasks.

These features are presented in Tables 3.1, 3.2 and 3.3 with respect to the main frameworks, tools, presented in the paper, and classified according to the three knowledge layers.

As shown in Table 3.1, data-driven approaches work on low-level data. These approaches mainly adopt linguistic, statistical and unsupervised methods, such as clustering, word-space model and subsumption relations. The knowledge representation is strongly based on term-to term relationship, often generating flat ensemble of terms. In some cases other formal methods, such as Formal Concept Analysis (FCA), fuzzy set or graph theory, are also used, that produce a kind of term structuring. Data-driven approaches do not exploit semantic annotation or require external ontology-based tools support. Although the work presented in [32] may seem an exception, it is a tentative to combine two different data spaces extracting from the same dataset, but describing terms and (semantic-oriented) RDF-tags, in order to mix the data with a different feature nature, in the clustering generation.

Table 3.2 shows information-driven approaches that usually extract relations between keywords or (more complex key-phrases) named entities adopting supervised or semi-supervised methods. The enhancement with respect to previous layer is that these approaches exploit topological representations for raw data extracted from text. Formal models, such as fuzzy set, part-whole relations and entity relations, induct richer semantic relations between named entities. The revealed tendency is to find patterns useful to group named entities and keywords and clustering methods are largely used in order to fulfill this task. Some approaches gather additional information from external tools, such as thesauri and knowledge bases, in order to better identify the terms sense and then produce more meaningful relations among NE and keywords. Table 3.2 hightlights that information-driven approaches are mainly used in Information Retrieval field.

Concept-driven approaches shown in Table 3.3 are aimed at producing a more refined representation of the corpus in input, achieving a knowledge structuring that evidences a deeper granularity of the information. These approaches focus mainly on building

a conceptual map or term-dependency network that allow the high-level knowledge description. At this purpose, they propose various methodologies based on formal models that reveal hierarchies or tree-based structures, such as Formal Concept Analysis (FCA), Conceptual Ontological Graph (COG); they exploit fuzzy modifiers to capture the vagueness in written text, that is hidden behind lexical relations and grammar dependencies; then they exploit these semantic and lexical connections to get the taxonomy and ontology learning. In general, the most of these approaches extract knowledge from external sources, but some of them build a conceptual structure based exclusively on the analysed text corpora.
External support, for these methods, includes both syntactical and semantic sources and tools. The most used external tool seems to be WordNet, whose synsets yield synonyms from each term sense, useful to contextualize concepts. Other external sources are knowledge bases, such as DBpedia, verb-lexicon (e.g. VerbNet[1]), as well as more sophisticated semantic tools, such as semantic frameworks (e.g. FrameNet[2]) and ontology-based knowledge organisation schema (SKOS[3]). Concept-driven approaches are used as well in Information Retrieval and Natural Language Processing areas.

Some approaches seem to prefer methodologies, capable of producing a topological conceptualisation of data, such as Formal Concept Analysis (FCA), Fuzzy sets, Graph Theory, etc. which are largely used to extract relations, patterns and recognize articulated concepts and topics. Since the Concept Mining approaches aim at generating more complex topological structures, they often combine methodologies and technologies from the three fields taken into consideration: NLP, IR, SW.

In conclusion, the Text Mining approaches presented in the literature combine methodologies from the three macro-fields and subfield (IR is mainly adopted in the information-driven ones). Topological semantic similarity measures are the most used simi-

[1] http://verbs.colorado.edu/ mpalmer/projects/verbnet.html

[2] https://framenet.icsi.berkeley.edu/fndrupal/

[3] https://www.w3.org/2004/02/skos/

Table 3.1: Data-driven approaches test results

Approach	**Research sub-fields**	**Knowledge representation**	**Ontology-based support**	**Similarity measure**	**Semantic annotation support**
Phillips [33]	NLP, clustering conceptual maps	syntagmatic lexical networks networks	n.a.	co-occurrence similarity	No
Schutze [34]	IR, EM clustering	Word space model	n.a.	Semantic similarity	No
Sanderson and Croft [35]	IR, subsumption relation	Subsumption relation-based hierarchy	n.a.	semantic similarity	No
Loia et al. [36]	SW, P-FCM	Word space model	n.a.	Proximity measure	Yes
Loia et al. [32]	SW, P-FCM, Collaborative clustering	Word and RDF-tag space model	n.a.	Proximity measure	Yes
Lau et al. [37]	NLP, POS tagging	Fuzzy subsumption relation hierarchy	n.a.	term frequency, mutual information	No
Kruschwitz [38]	IR	HTML tag structure	n.a.	semantic similarity	No
Chuang and Chieng [39]	IR, agglomerative clustering	suffix tree hierarchy	n.a.	topological measure	No
Baeza-Yates [40]	IR, graph theory	graph-based relations	n.a.	semantic measure	No

larity measures, since they are more effective than frequency-based and statistical measures to extract relations between terms or concepts. Although data-driven approaches are based on term frequency-based measures, such as co-occurrence measure, tf-idf, mutual information to assign a weight to each word, information and concept-driven approaches employ semantic similarity, especially topological measures, in order to better represent complex relationships among articulated concepts. The most used measure acts on lexical similarity to extract hyponymy, hypernymy and WordNet synsets, as well as fuzzy measures, which provide a more sensible evaluation of the ambiguousness about NE and concepts.

The use of external sources seems more useful when dealing with high-level input data, i.e., concepts, or when the modelling requires higher-level conceptualisations. The semantic annotation support is instead frequent in the concept-driven and information-driven approaches, while it is almost missing with data-driven approaches (see Table 3.1). The data-driven approaches indeed work mainly on row data that generate not very complex knowledge (often composed of terms ensemble), so the semantic annotation

Table 3.2: Information-driven approaches test results

Approach	Research sub-fields	Knowledge representation	Ontology-based support	Similarity measure	Semantic annotation support
Girju et al. [41]	IR Classification rule extraction	Part-whole relations	n.a.	topological similarity	No
Snow et al. [42]	IR Classification	hyponymy hypernymy	WordNet	Taxonomy	Yes
Reichartz et al. [43]	IR, kernel methods	Parse tree	n.a.	phrase similarity	No
Giuliano et al. [44]	IR, kernel methods	entity relations (Synsets, Hypernym)	WordNet	Semantic similarity, WordNet synsets	Yes
Cao et al. [45]	IR, hierarchical fuzzy clustering	Fuzzy vector space model	n.a.	Fuzzy similarity measure	No
Diaz-Valenzuela et al. [46]	IR, document clustering, partitional clustering	clustering-based constraints	n.a.	Frequency measure	No
Mintz et al. [47]	IR relation extraction classification	Freebase relations	Freebase	entity-relation model	Yes
Loia et al.[48]	IR, topic extraction proximity-based fuzzy clustering	Fuzzy multiset	n.a.	Fuzzy measure	Yes

tools are not required at this stage. The use of external ontology-based tools is instead predominant when the knowledge becomes articulated and generates specialized conceptualization. Table 3.3 indeed provides a list of concept-driven approaches that reinforce the generated knowledge with the support of external semantic sources and databases.

The reminder of this chapter focuses on methods to extract knowledge from texts (3.2) and multimedia data (3.3).

3.2 Knowledge extraction from texts...

3.2.1 Concept Mining

Current research trends are interested in knowledge acquisition from text, especially in Concept Mining, whose goal is the extraction of concepts from artifacts. Concept Mining represents a subfield of Text Mining, aimed at extracting concepts from text.

Table 3.3: Concept-driven approaches test results

Approach	**Research sub-fields**	**Knowledge representation**	**Ontology-based support**	**Similarity measure**	**Semantic annotation support**
Della Rocca et al. [49]	IR, Conceptual analysis	LDA-based concept learning	SKOS, WordNet	statistical (LSA)	Yes
Cimiano et al. [50]	IR, Formal Concept Analysis (FCA)	FCA-based hierarchy	n.a.	Frequency measure, topological measure	No
De Maio et al. [51]	IR, Conceptual analysis	Fuzzy FCA-based hierarchy	DBpedia, WordNet	hierarchical	Yes
Loia et al. [52]	Sentiment Analysis, Sentic Computing	Fuzzy modifiers, fuzzy sets	WordNet, SentiWordNet	Semantic similarity, WordNet synsets	Yes
Ontolearn [53]	NLP, statistical statistical	Taxonomy learning	WordNet, FrameNet, VerbNet	Topological similarity	Yes
Navigli et al. [54]	NLP, statistical approaches	Taxonomy learning,	WordNet, Dmoz	Topological similarity, synsets	No
Valarakos et al. [55]	NLP, Knowledge discovery	Ontology learning, HMM	n.a.	Statistical similarity	Yes
Alani et al. [56]	IR, knowledge extraction	Syntactic analysis, semantic analysis	Gate, WordNet WordNet	Semantic similarity similarity	Yes
Shehata et al. [57]	IR, concept extraction	Conceptual ontological graph (COG)	n.a.	Topological similarity	No
Agirre et al. [58]	NLP, word-sense disambiguation	topic signature WordNet concepts	WordNet	Topological similarity	No

Concept Mining approaches are largely used in many fields, such as Information Retrieval (IR) [59] and Natural Language Processing (NLP) [60]. The main applications include detecting indexing similar documents in large corpora, as well as clustering document by topic. Most of the common techniques in this area are mainly based on the statistical analysis of term frequency, to capture the relevance of a term within a document [61]. More accurate methods need to capture the semantics beyond the terms. Traditionally, concept extraction methods employed thesauri, such as WordNet[4], to transform words extracted from documents in concepts. The main issue related to thesauri is that the word mapping to concepts is often ambiguous. Ambiguous terms can be generally related to more than one concept, only the human abilities allow contextu-

[4] https://wordnet.princeton.edu/

alizing terms and find the right concept to which terms belong. Since thesauri do not describe the context along with the terms, further techniques have been introduced to face the word sense disambiguation; some of them perform linguistic analysis of text, based on term frequency similarity measures, some others employ knowledge-based models to generate a context useful to evaluate a semantic similarity between concepts.

Since documents are often described as a sequence of terms, a widely used data representation adopted by many methods is the vector space model (VSM) [62, 63, 64]. The VSM model represents each document as a feature vector of terms (words and/or phrases) present in the document. The vector usually contains weights of the document terms, defined mainly on the frequency with which the term appears in the document. Other techniques extend the VSM representation with context-related information for each term, transforming the VSM vector in a global text context vector. This way, a global context about a term is built by merging its local contexts, which are derived from each document where the term appears in [65].

Similarity between documents is calculated by similarity measures, which evaluate the document similarity as the similarity between their feature vectors. Similarity measures are various, the most used include euclidean, cosine, Jaccard, etc. Other techniques use document attribute information involved in query and possessed jointly by documents to evaluate, in order to extract inter-document information useful to calculate their similarity. These techniques are called query-sensitive similarity [66].

In Text Mining domain, the term importance in a document is based on the frequency with which the term appears in the document. Moreover, a high frequency does not mean that a term contributes more to the meaning of a sentence. There are some words with a low frequency which provide key concepts to the sentence. The importance of the term is important as some summarization techniques, which represent their summarizes based on the importance of their words [67]. The sentence meaning usually depends strongly on verbs and their arguments. Verb analysis allows finding out who is doing something, or acting toward something or someone else, clarifying each term role in explaining the meaning of the sentence topic.

Whether extracting sophisticated information or simple ones, these techniques constitute the underlying methodological background of the Concept Mining research area and most approaches are modeled and developed on the basis of them.

In this section, an approach for concept detection is discussed. This approach discriminates straightforward concepts from a document corpus, through the discovery of their context, i.e., the surrounding text (words or sentences) that describes the concepts.

3.2.2 Simplicial complex model

The concept extraction from text lies on a straightforward formal model, the simplicial complex. The approach builds a topological space called Simplicial Complex, which connects points composing incremental geometrical structures, such as line segments, triangles and their n-dimensional counterparts. The most elementary structure that constitutes the complex is a simplex S, defined as follows:

Definition 1. *n-Simplex. A semantic n-Simplex or simple n-Simplex S is a set of independent abstract vertices* $(v_0, v_1, ..., v_n)$ *that constitutes a convex hull of* $n + 1$ *points.*

For example, a 0-Simplex is a singleton representing a vertex (e.g. the set (v_1) where v_1 is a vertex), a 1-Simplex is a two elements set that corresponds to an open segment (v_1, v_2), a 2-Simplex is a three elements set representing an open triangle that does not include its edges and vertices (v_1, v_2, v_3). Generally, an n-simplex is the high dimensional analogy of those low dimensional simplexes (segment, triangle, tetrahedron and so on) in n-space. Geometrically, an n-simplex uniquely determines a set of linearly independent vertices and vice versa. It is the smallest convex set in a euclidean space R^n that contains $n + 1$ points $v_0, ..., v_n$ that do not lie in a hyperplane of dimension less than n.

The n-simplex is a basic structure but it can be formed by more elemental structures called faces.

Definition 2. *r-Face*

Given an n-Simplex $[v_0, v_1, ..., v_n]$*, an r-face is an r-Simplex* $[v_{j0}, v_{j1}, ..., v_{jr}]$ *whose vertices are a subset of* $(v_0, v_1, ..., v_n)$ *with cardinality* $r + 1$*.*

The convex hull of any r vertices subset of the n-simplex is an r-face. The 0-face, 1-face, 2-face for example, are respectively points, edges and triangles of the n-simplex, whereas the n-face is the whole n-simplex. Then, the simplicial complex may be defined as an overall structure, composed of one or more simplexes; this complete geometrical structure is suitable to represent terms and their relationships in our approach.

Definition 3. *The simplicial n-complex*

The simplicial n-complex C *is a finite set of simplexes that satisfies the following two conditions:*

- *Any face of a simplex from* C *is also in* C
- *The intersection of any two simplexes* $S_1, S_2 \in C$ *is a face of both* S_1 *and* S_2*.*

In other words, an n-complex is a closed set of m-simplexes, with $m \leq n$ and n is the maximal dimension of a simplex in the n-complex. The union of the vertices $v_0, v_1, ..$ of all the m-simplexes are the vertices of the n-complex and all h-faces of the simplexes are also contained in the complex. An example: if the 3-Simplex $\{a, b, c\}$ is in the n-complex C then its r-faces $\{a, b\}$, $\{b, c\}$, $\{a, c\}$, $\{a\}$, $\{b\}$ and $\{c\}$ belong to C.

In order to properly describe the relationships between terms in our modelling, other notions about a complex structure will be introduced as follows.

Definition 4. *Direct connection between simplexes*

Two simplexes in a n-complex are directly connected if the intersection among them produces a non empty h-face with $h \leq n$*.*

Thus, if $A = \{a, b, c\}$ and $B = \{b, c, d\}$ are two simplexes in a n-complex and their intersection is the non empty set $\{b, c\}$, then they can be said *directly connected.*
In general, two non empty simplexes are h-connected or simply connected if a finite sequence of directly connected simplexes connecting them exists. More formally,

Definition 5. *h-Connection of simplexes*
Let $A = S_1, S_2, \ldots, S_m = B$ be non empty simplexes, then, A and B are h-connected if every pair of consecutive S_i and S_{i+1} has a h-face in common with $i = 0, 1, 2, ..., m-1$ where $h \leq n$.

For instance, if $A = \{a, b, c\}$, $B = \{b, c, d\}$ and $C = \{d, e, f\}$ are simplexes in an n-complex, then (A, B) and (B, C) are two directly connected pairs of simplexes by the 1-face $\{(b, c)\}$ and the 0-face $\{(d)\}$ respectively (see Figure 3.3). This implies that (A, C) are two connected simplexes because of (B) which is directly connected with both (A) and (C).

Definition 6. *(n, k)-skeleton*
An (n, k)-skeleton is the n-complex in which all the m-simplexes, of dimension m, with $m \leq k$, and their faces have been removed.

Notions about connected simplexes and skeletons are strictly related. The h-connected component in a skeleton may be defined as the maximal h-connected subcomplex (it is in turn, a complex composed of h-connected simplexes) of an n-complex, which implies that does not exist any other h-connected component that is superset of it.

As an example, let us consider the complex built on the three simplexes, mentioned above, A, B and C (Figure 3.3 a), all the conceivable faces are: $\{(a, b)\}$, $\{(b, c)\}$, $\{(a, c)\}$, $\{(c, d)\}$, $\{(b, d)\}$, $\{(d, e)\}$, $\{(d, f)\}$, $\{(e, f)\}$, $\{(a)\}$, $\{(b)\}$, $\{(c)\}$, $\{(d)\}$, $\{(e)\}$, $\{(f)\}$. With k=0, the (3,0)-skeleton is the original n-complex with the three main simplexes and all their faces. A (3, 1)-skeleton can be built by removing all its own 0-simplexes (see Figure 3.3 b); the

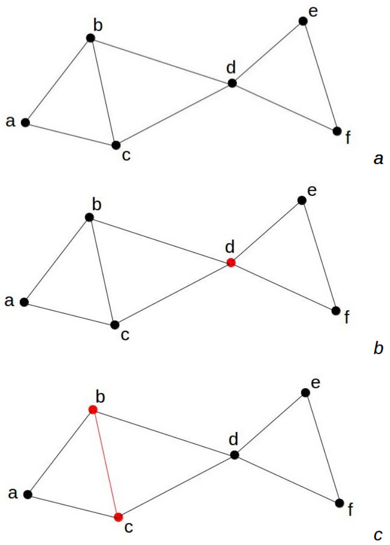


Figure 3.3: An example of skeleton

new complex is composed of the three simplexes A, B, C and their edges are: $\{(a,b)\}$, $\{(b,c)\}$, $\{(a,c)\}$, $\{(c,d)\}$, $\{(b,d)\}$, $\{(d,e)\}$, $\{(d,f)\}$, $\{(e,f)\}$. According to Definition 4 and Definition 5, the simplexes A and B are still connected, while simplexes B-C, and A-C not yet because the face (d) has been removed. The removal of all the 2-Simplexes from the previous structure ($k = 2$), and particularly $\{(b,c)\}$ (Figure 3.3 c), produces the (3, 2)-skeleton, that is a new complex composed of the three simplexes A, B, C, which are not connected between them because their intersections are empty sets. This structure is very useful to analyse vertices and the relationships between them, considering different levels of detail.

3.2.3 Simplicial complex of terms for concept extraction

The simplicial complex represents a topological space suitable to describe linguistic relations among terms extracted from the text

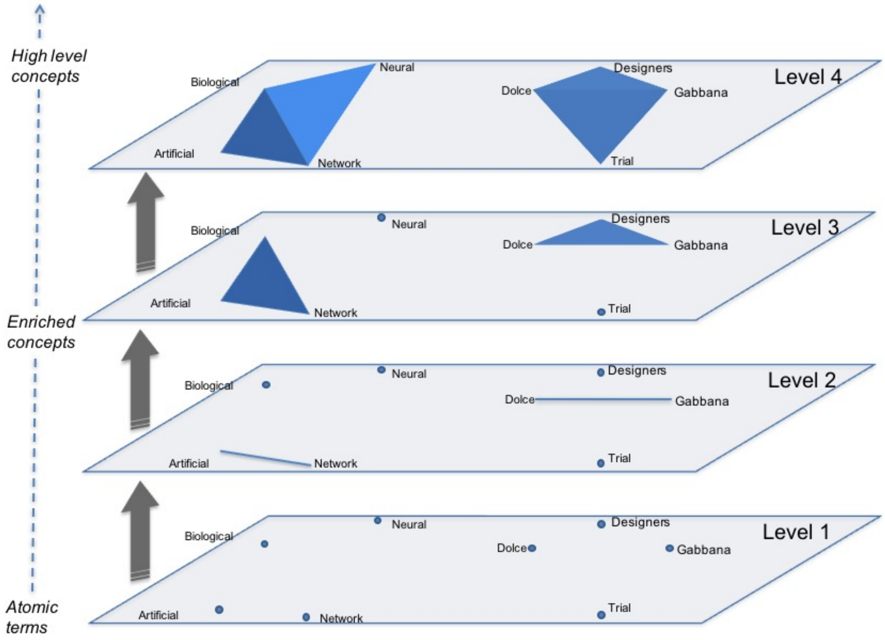


Figure 3.4: A concept-based perspective of a complex: level by level, terms are extended by distance-based connections, generating specific concepts.

corpora. In fact, each term can be represented by a vertex of the structure, the edges between the vertices represent the relationships between these terms. This way, simplexes of terms can represent concepts. Simplexes, aggregated through various skeletons, represent more precise and contextualized concepts. Figure 3.4 shows a level-based representation of terms and their relationships. Each level shows a different conceptualization depending on how the terms are linked to each other. A 0-simplex, for example, is the term *Network* which expresses the generic concept associated with its own term label. This 0-simplex can be combined with other 0-simplexes (i.e., concepts) to form the following three 1-simplexes: (computer, network), (traffic, network), (neural, network). These sets express more detailed and specific concepts than the individual 0-simplexes, hence two 2-simplexes are built by starting from

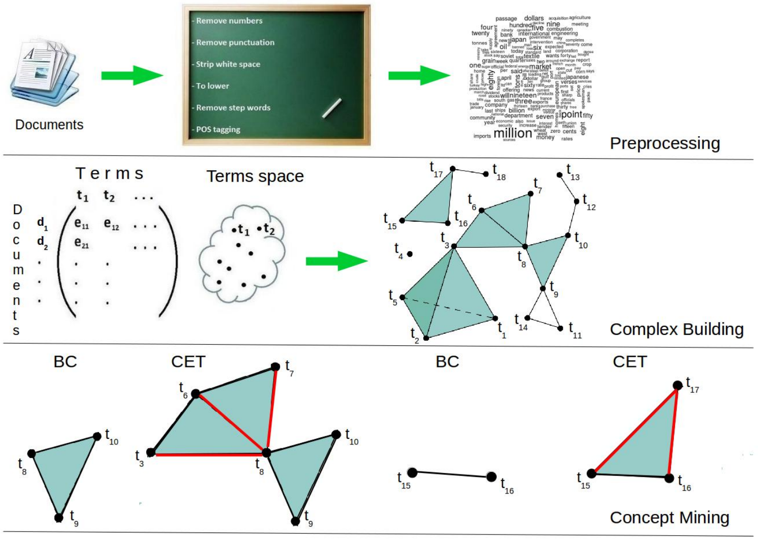


Figure 3.5: A logical overview of the framework

1-simplexes: (artificial, neural, network) and (biological, neural, network) express a richer semantics and describe a more precise concept. This layered contextualization is the key for disambiguation, which leads to robust concept extraction. In fact, the incomplete and generic concept *neural network* could be properly characterized by the term *artificial* defining the very specialistic concept *artificial neural network*.

The simplicial complex construction has been used to define a framework for concept extraction from texts. Figure 3.5 shows the main steps of the approach, divided into three macro phases: *Preprocessing*, *Complex Building* and *Concept Mining*. In the *Preprocessing* phase, the input document corpus is processed to remove the "noisy" text (numbers, punctuation, stop words, etc.): only terms that represent nouns and adjectives in the grammatical categories are selected for the next phase.

The *Complex Building* phase builds the simplicial complex as skeletons representing different conceptualizations, which range from a more basic view to a more general and cross-topic one. To achieve this, the relevance of the extracted and pre-processed terms, with respect to the document corpus, is assessed by using the *tf-idf* (term frequency-inverse document frequency) measure [68]. Then, given a corpus composed of n documents, each term t is expressed as an n-dimensional column vector $(v_1(t), v_2(t), ..., v_n(t))$, such that the i-th value $v_i(t)$ (evaluated by the tf-idf measure) represents the relevance of t in the i-th document of the corpus. The vectors from the terms form a document-term matrix. To build the matrix, terms are selected by comparing their tf-idf value with a given threshold $FeaTHR$. This term selection allows to get accurate concepts and speed up the complex building. Relationships between terms (column vectors) are assessed through the euclidean distance. Since term vectors are normalized to one with respect to the Euclidean norm, the Euclidean distance is equivalent to the cosine measure [69]. According to term relationships, the structure of the simplicial complex is incrementally built by connecting terms on several layers. At each layer k, if Definitions 1 - 5 hold, an (n, k)-skeleton is built by connecting simplexes according to Definition 6. The process starts from all 0-simplexes to gradually reach more connected components by increasing the k level. In order to preserve the complex definitions, the size of each simplex must not be greater than the number of levels reached till that moment. More formally:

Definition 7. ***Simplexes size upper bound (SSUB)*** *At the k-th level, an (m-1)-simplex can be extended with a new term t_j, if a term t_i exists in the simplex, such that*

$$\underset{t_j}{\operatorname{argmin}}\ d(t_i, t_j) \tag{3.1}$$

Then, the new m-simplex contains j terms, with $j \leq k$.

This definition outlines the principal property of the simplicial complex (Definition 3), which asserts that in an n-complex the highest sized simplexes can not contain more than $n + 1$ vertices (terms). This upper bound guarantees that the building of simplexes prevents their fast expansions that can generate interleaving concepts. At each level indeed, the simplexes with the highest size can add only one new term to themselves, thus only new terms highly related are included in simplexes. On the other hand, a disadvantage of this definition is to get simplexes whose vertices (terms) are very far to each other. The following definition prevents this effect:

Definition 8. ***Weak relationships cut off radius per term (WRCRT)***

Given z m-simplexes $S_1^k, S_2^k, ...S_z^k$*, at the k-th level, a term* t_j *is a candidate to be part of the m-simplex* S_i^k *if* $\forall t_i \in S_i^k$

$$d(t_i, t_j) \leq \tau \tag{3.2}$$

where τ *is an a-priori fixed threshold, with* $i, j \leq m$ *and* $i \neq j$*.*

WRCRT is a covering radius, which considers only relationships among terms that fall into their own neighbourhood. This threshold constraints the relationships useful to link simplexes, and thus, to connect concepts. For this reason, fixing small values of WRCRT allows to cut off the weakest relationships per each term selecting the most specific concepts. On the contrary, big values of WRCRT allows to consider very general (cross-topic) concepts.

Algorithm 1 implements the whole structure building process. The algorithm takes in input the document-term matrix M, euclidean distance matrix among terms D, the WRCRT threshold τ and the maximum number of levels K_{max}. At each level k, new simplexes $s_1^k, s_2^k, ..., s_z^k$ are generated or unified according to the (n, k)-skeleton property. Lines 5-13 trace all the evolution of the simplexes and a data structure S stores the simplexes collection produced at each iteration k. At each iteration k (until to K_{max}), a new level is explored, i.e., distance-based relations are considered to unify simplexes, whereas simplexes with size greater than $k+1$ at iteration k are filtered out, and will be reconsidered from the level $k+1$. For each term pair t_i and t_j, the distance $d\left(t_i, t_j\right)$ in D between them is evaluated in order to decide if the simplexes, which t_i and t_j belong to, must be connected or not in the current skeleton S^k. Line 6 condenses the definition conditions about SSUB and WRCRT (Definitions 7 and 8). The algorithm output is a skeleton collection S, where each skeleton S^k is composed of a simplex list representing the concepts.

The *Concept Mining* phase is in charge of discovering admissible concepts. Two types of concepts are extracted: the Basic Concepts (BCs) and the Concept Extending Terms (CETs). The former relate to specific concepts composed of terms strongly related to each other (i.e.,"Dolce & Gabbana"). The latter are BCs extended with terms which define the context extending the initial conceptualization of the BC (i.e. CETs of "Dolce & Gabbana" can be "fashion", "designers", etc.).

Algorithm 1 Build a skeleton collection of terms

```
Require: $M$: $n \times m$ document-term matrix, whose terms are $t_1, t_2, ..., t_m$
    $D$: ascendant ordered distance $D = \{d(t_i, t_j) \mid i \neq j, i, j = 1, .., m\}$
    $\tau$: prefixed WRCRT
    $K_{max}$: prefixed maximum number of levels
Ensure: $S$: list of skeletons of the simplicial complex of terms
$k = 0$
$S = \emptyset$
while $k \leq K_{max}$ do
    Let $t_i, t_j$ be two terms in $M$ and $s^k_i$, $s^k_j$ the simplexes which respectively they belong
    to (s.t. $s^k_i \cap s^k_j = \varnothing$).
    for all $d(t_i, t_j)$ in $D$ do
        if $d(t_i, t_j) \leq \tau \wedge \left| s^k_i \cup s^k_j \right| \leq k + 1$ then
            $s^k(i, j) \leftarrow s^k_i \cup s^k_j$
            Add $s^k(i, j)$ to $S^k$
        end if
    end for
    Add $S^k$ to $S$
    $k \leftarrow k + 1$
end while
return $S$
```

In order to extract BCs and CETs, the complex building process described in Algorithm 1 has been executed several times on different values of WRCT τ to get the most lasting simplexes through the levels, which are the simplexes candidate to represent BCs, formally:

Definition 9. ***BCs extraction through threshold on levels***
Given a prefixed threshold $extTHR$, a simplex s of a simplicial

complex is a basic concept (BC) if:

$$BC\,(s) = \begin{cases} 1 & \textit{if } mean\,(s) > extTHR \\ 0 & \textit{else} \end{cases} \tag{3.3}$$

where mean(s) is the mean number of levels of s, in which s rests unchanged, computed on level sums per run $l\,(s,\tau) = \sum_{k=0}^{Kmax} s^k$*, with* $s^k = s^k - 1$ *by varying the WRCRT* τ *in the range [sd , fd] with incremental step inc:*

$$mean\,(s) = \frac{\sum_{\tau=sd}^{fd} l(s,\tau)}{Rmax} \tag{3.4}$$

The number Rmax of possible runs of Algorithm 1 trivially is $Rmax = ((fd - sd)/inc) + 1$.

Once all the BCs have been identified according to Definition 9, all the other terms added to the BCs by incrementing the τ value are considered as CETs. As stated, a CET represents wide concepts that extend a BC. Let us notice that the (distance-based) ordering to add new terms in CETs suggests the relevance of that new term in enriching the primary concept related to the BC, thus, as additional terms are connected to an initial BC, new, extended (often cross-topic) concepts are identified. The final result is a list of BCs with the related CETs as two distinct associated simplexes. Then $ExtTHR$ represents a threshold for the concept extraction, assessed as the mean of levels on all of the runs. It judges whether a linkage among terms is strong enough to represent a BC. A high $ExtTHR$ value guarantees BCs formed by strong relations (but often they are few); a low $ExtTHR$ captures more BCs, which could be formed by terms that are not strongly connected. The choice of the ExtTHR value strongly depends on the selected dataset and it is chosen during the experimental stage.

In order to clarify the comprehensive procedure, a sketched example is given as follows. Let us suppose the initial configuration: $sd = 0.1$, $fd = 0.8$, $inc = 0.1$. According to the description of the approach, Algorithm 1 is executed ($Rmax = 8$) eight times, by varying the parameter τ in [0.1, 0.8]. Listing 3.1 shows the construction of some simple simplexes, considering the two initial terms (1-Simplexes) "Dolce" and "Gabbana" whose distance satisfies the relation in line 6 of Algorithm 1. Starting with $\tau = 0.1$, a 2-Simplex {Dolce, Gabbana} is built, connecting the two terms. With the next τ values, no additional edges are added, because no distance between two terms is lower or equal to τ; this holds for the next runs of Algorithm 1, respectively with $\tau = 0.1, 0.2, 0.3, 0.4, 0.5$. Listing 3.1 sketches only the runs of Algorithm 1 with $\tau = 0.1$ and $\tau = 0.5$, where the only simplex is {Dolce, Gabbana} which does not change during these executions.

```
τ = 0.1

1−Simplex(Dolce),
1−Simplex(Gabbana)             Level 1


2−Simplex(Dolce, Gabbana)      Final Level

τ = 0.5

1−Simplex(Dolce),
1−Simplex(Gabbana)             Level 1


2−Simplex(Dolce, Gabbana)      Final Level
```

Listing 3.1: BCs extraction, with $\tau = 0.1$ and $\tau = 0.5$

```
τ = 0.6

1−Simplex(Dolce),
1−Simplex(Gabbana)                  Level 1


6−Simplex(Dolce, Gabbana,
Designers, Trial, Defense,
Prosecutor)                         Level 6


13−Simplex(Dolce, Gabbana,
Designers, Trial, Legal, Defense,
Prosecutor, Risky,
Dangerous, Construction, Work,
Child, Father)                      Final Level
```

Listing 3.2: BCs extraction with $\tau = 0.6$

```
τ = 0.7

13−Simplex(Dolce , Gabbana,
Designers ,Trial , Court ,
Course , Weaky, Idea ,
Problems , Attorney , Legal ,
January , Short)                    Final Level

τ = 0.8

20−Simplex(Dolce , Gabbana,
 Designers , Trial , Court ,
Investigation , Safety , Miles ,
Spokesman , Xinhua , Crossing ,
Garcia , Childish , Parents , Son,
Strategic , Western , Forces ,
Libyan , Rebels)                    Final Level
```

Listing 3.3: BCs extraction with τ= 0.7 and τ= 0.8

Listing 3.2 shows the evolution of previous simplex with $\tau = 0.6$. Level by level, the Algorithm 1 tries to find the term connections. The simplex at 1-st level was previously analyzed. At 6-th level, four new terms are added to the simplex (Designers , Trial , Defense, Prosecutor) that enrich the previous simplex, while at final level, there is a simplex compound of new further words, {Dolce, Gabbana, Designers, Trial, Legal, Defense, Prosecutor, Risky, Dangerous, Construction, Work, Child, Father}.
Increasing τ, i.e., increasing the covering distance, other terms can be added, as shown in Listing 3.3, where, with $\tau = 0.7$, the simplex is composed of a common subset of the previous analyses and some new more specific terms such as "Court", "Attorney", etc. Similar analysis happens with $\tau = 0.8$.
Generally, the last terms added are more general, cross-topic and based on weaker relationships, for this reason they can easily change in the last distinct runs.
Let us suppose that ExtTHR = 8, Rmax = ((0.8 − 0.1) /0.1)+1 = 8 then, according to Definition 9, the terms "Dolce", "Gabbana" will be compound a BC: $mean(\{Dolce, Gabbana\}) = (20+20+20+18+12+6+5+3)/8 = 13 > ExtTHR$, where each addend in the sum is the number of levels where the 2-Simplex("Dolce", "Gabbana") rests unchanged, through a run. Thus, the 2-Simplex("Dolce",

"Gabbana") is a BC("Dolce", "Gabbana").
Then, the simplexes built from this BC (by increasing τ), adding new terms will identify complete concepts about the BC("Dolce", "Gabbana"). For example the term "designers" refers directly to the BC "Dolce & Gabbana", the famous fashion designers, while terms like "trial", "court", "investigation", "safety", "prosecutor", "legal", "defense", "miles", "spokesman", "course", etc. describe a fact about a trial on a copyright infringement that involves Dolce & Gabbana. Increasing τ leads to a further generalization of the topic "trial" that involves also a trial relative to prisoners of libyan war, represented by terms like "western", "forces", "libyan", "rebels", etc. that refer to another BC.
Let notice that a wider generalization (built with simplexes compound by many levels, i.e., by increasing τ) could introduce too terms that ambiguously would describe a concept.

3.2.4 Approach evaluation

Our experimental evaluation aims to assess the effectiveness of our approach in identifying accurate conceptualization (in terms of BCs and CETs).

The experimentation was conducted on two datasets: 500N-KPCrowd-v1.1 and Reuters-21578[5]. The former contains 500 documents, 72,713 words in total after preprocessing, divided into ten topic categories. The latter, subset of the well-known Reuters-21578, contains 16,368 after the preprocessing phase. The approach quality in identifying concepts has been tested by comparing simplicial complex results to results of two baseline methods. The first one is AlchemyAPI, which is commercial tool for text mining including a set of NLP features (i.e., named-entity extraction, sentiment analysis, etc.). The second method consists in the manual annotation of relevant terms, keyphrases, compound terms, present in documents, from the two datasets 500N-KPCrowd-v1.1 and Reuters-21578.

[5] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Two main experiment types have been conducted: one is oriented to extract the basic concepts (BCs) the other extends them to get richer and wide concepts, i.e., the concept expanding terms (CETs).

The simplicial complex approach builds BCs from the scratch and they are not a-priori classified or labeled. Each BC is compared with concepts from the baseline. If a match occurs, the BC can be of two different types:

- Perfect Matching BC: the extracted BC is equal to the concept from the baseline i.e., it is composed of a set of words that is exactly the same occurring in the reference concept.
- Partial Matching BC: the extracted BC is partially equal to the concept from baseline, i.e., a subset of words matches at least half of words in the reference concept.

Before presenting the tests on BCs, an initial study has been achieved to adequately tune the parameters of the simplicial complex approach, and to show the overall system performance. The experiments have been carried out on the *500N-KPCrowd-v1.1* dataset, using AlchemyAPI as baseline method. Table 3.4 shows the precision and recall, with different parameter configurations. According to Definition 9, the incremental setting in range [0.1, 0.8] with step=0.1 is fixed, with the maximum number of levels to generate, K_{max}= 20.

The parameters taken into account are the threshold $FeaTHR$, for selecting the most meaningful terms used in the experiment, and the extraction threshold *extTHR*, that affects the concept extraction (see Definition 9). Precision and recall are calculated on the BCs extracted by our framework with respect to the concepts, i.e., named entities (NE) returned by AlchemyAPI; precisely, precision and recall are evaluated on the Perfect Matching BC (PeM), i.e. perfect matches between BC and NE, and on Partial Matching BC (PaM), i.e., partial matches between the extracted BC and the reference NE.

Table 3.4 evidences that increasing the number of terms (i.e., by reducing the $FeaTHR$ value), involved in the building of the

Table 3.4: Precision and recall on *500N-KPCrowd-v1.1* dataset w.r.t. all the AlchemyAPI categories (with sd= 0.1, fd=0.8, inc=0.1, kmax=20)

FeaTHR	**ExtTHR**	**Precision**		**Recall**	
		PaM	**PeM**	**PaM**	**PeM**
0.4	10	12.76	22.8	15.83	20.94
0.4	7	26.25	27.44	25.27	24.94
0.4	5	34.36	31.72	31.21	27.55
0.4	3	16.57	11.82	22.68	8.01
0.3	10	32.84	27.68	29.68	33.21
0.3	7	34.73	30.39	30.15	28.59
0.3	5	37.58	33.76	34.55	28.24
0.3	3	29.37	26.53	24.95	11.24
0.2	10	32.29	33.59	28.51	36.57
0.2	7	35.74	38.24	31.43	35.25
0.2	5	47.36	44.18	41.98	38.90
0.2	3	32.36	35.15	27.34	14.78
0.18	10	32.21	37.12	26.87	33.20
0.18	7	30.12	31.52	25.67	31.66
0.18	5	48.93	35.88	43.73	28.24
0.18	3	37.47	33.87	28.91	14.85
0.12	10	33.83	35.83	32.71	21.16
0.12	7	41.83	31.62	36.62	27.71
0.12	5	57.66	41,27	52.11	41.83
0.12	3	36.48	29.27	30.63	20.13

complex, contributes to improve precision and recall results, along with the extraction threshold ($ExtTHR$) value which affects the construction of BCs: it has to be not too small because it can select many BCs composed of single terms, but at the same time, too high ExtTHR values can cut off terms for composing BCs, leading to more cross-topic and confused BCs.

Let us notice that by varying the FeaTHR value, the best recall and precision values for PaM and PeM are generally with $ExtTHR = 5$, evidencing that, with this dataset, strong linkages among terms form relevant concepts when those terms stay connected in about 6 levels, for each run, according to Definition 9. The best result is with $feaTHR$ equals to 0.12 and $ExtTHR$ equals to 5, as shown in Table 3.4, where the recall on PeM is greater than 40% and the recall on PaM overcomes 50%, precision on the perfect matches also overcomes the 40%, while it is almost 60% on the partial matches. The performance is comprehensively satisfactory, considering that the dataset is human generated, and so the selection of keyphrases can result more accurate and precise, enriched with extra words that could not be included in the given documents. Let us recall that $feaTHR$ equals to 0.12 means that almost all the terms extracted in the preprocessing are candidate to form the feature space (in this case, only the 7.2% of terms are discarded).

Let us notice that when FeaTHR decreases, see Table 3.4, the framework produces increasing total recall and precision, even though the PeM tends to decrease on the highly dimensioned feature sets. As an example, with FeaTHR = 0.2 and ExtTHR = 5, the recall on PaM is 41.98% and on PeM is 38.90%, while the experiment with FeaTHR = 0.18 and ExtTHR = 5 the PaM for recall improves (43.73%), because more terms contribute to define the BCs but, at the same time, the same terms affect the value of the perfect matches (PeM) that decreases (28.24%).

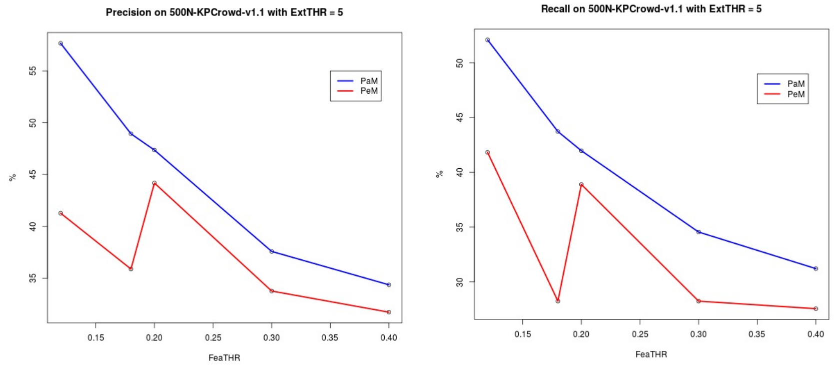


Figure 3.6: Precision and recall on 500N-KPCrowd-v1.1, by varying the number of features (terms)

Figure 3.6 shows the tendency of PaM and PeM for recall and precision, by varying FeaTHR, with ExtTHR = 5. As stated, in general, increasing the features set involved in the process, PaM and PeM assume higher values. On the contrary, reducing the features set, both their values decrease. One of the best values in terms of recall and precision are indeed given with ExtTHR =

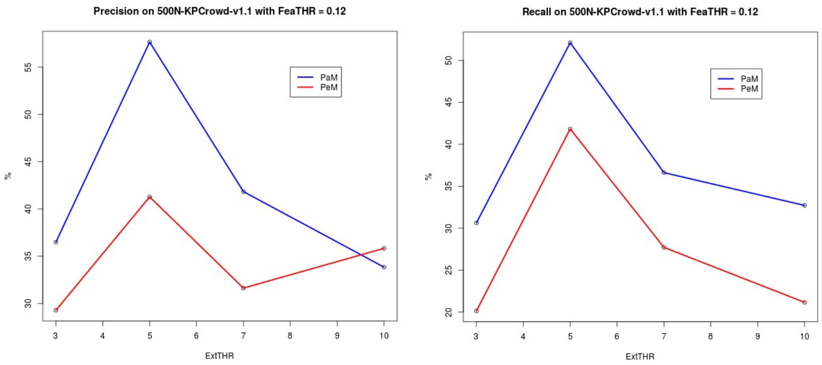


Figure 3.7: Precision and recall on 500N-KPCrowd-v1.1 by varying the extraction threshold $ExtTHR$

5, fixing FeaTHR=0.12. Figure 3.7 shows the tendency of PaM and PeM, with FeaTHR=0.12, by varying the extraction threshold ExtTHR. The ExtTHR threshold guarantees the selection of strong BCs (viz., BCs generated by simplexes that stay unchanged, given that threshold) with values equal to 5: the perfect matches on extracted BCs might keep increasing with values greater than 5 (i.e., there are further BCs that perfectly match the AlchemyAPI NEs); on the contrary, ExtTHR values greater than 5, enlarge concepts too much, degrading the number of partial matches.

In the light of the overall system performance analysis, the parameter setting with $feaTHR = 0.12$, and $ExtTHR = 5$ in Table 3.4 has been considered to accomplished the effective experimentation on BCs.

For each AlchemyAPI category, Table 3.5 shows three values for the precision and recall: the values of perfect and partial matches and their comprehensive value TOT. Let us notice that our approach accurately extracts BCs which can cover several categories of AlchemyAPI NEs: good precision and recall values are given for categories as Continent, EntertainmentAward, FieldTermnology, Natural Disaster, Organization and PrintMedia, especially in terms of total values (TOT). Lower recall appears with Country and Crime, even though they present quite good recall values on the

Table 3.5: 500N-KPCrowd-v1.1 dataset: precision and recall on BCs w.r.t. AlchemyApi NE for different categories

Category	**Precision**			**Recall**		
	PeM	**PaM**	**TOT**	**PeM**	**PaM**	**TOT**
Automobile	90.25	9.75	**100**	33.33	33.33	66.66
Company	15.46	43.04	58.50	13.95	38.83	52.79
Continent	**100**	0	**100**	**57.14**	28.57	**85.71**
Country	41.12	17.20	58.32	41.17	5.88	47.05
Crime	95.46	4.54	**100**	37.50	6.25	43.75
Degree	98,32	1,69	**100**	25.0	25.0	50.0
Drug	**100**	0	**100**	**50.0**	8.33	58.33
EntertainmentAward	**100**	0	**100**	28.57	57.14	**85.71**
Facility	0.0	89.67	89.67	0.0	79.88	79.88
FieldTerminology	1.06	98.94	**100**	1.88	88.36	**90.25**
GeographicFeature	20.82	79.18	**100**	2.27	68.18	70.45
HealthCondition	52	32	84	41.93	25.80	67.74
Holiday	**100**	0	**100**	16.66	33.33	50.0
JobTitle	25.24	65.04	90.29	21.66	55.83	77.5
Movie	86.27	6.48	92.75	16.66	50.0	66.66
NaturalDisaster	**100**	0	**100**	**100**	0.0	**100**
OperatingSystem	25.24	65.05	90.29	**50.0**	16.66	66.66
Organization	16.70	70.67	87.37	15.38	65.10	**80.48**
Person	7.62	54.45	62.07	6.95	49.71	56.67
PrintMedia	9.80	90.2	**100**	6.84	84.93	**91.78**
Region	**100**	0	**100**	10.52	68.42	78.94
Sport	**100**	0	**100**	**71.42**	0.0	71.42
StateOrCounty	33.76	28.57	62.33	26.53	22.44	48.97
Technology	91.76	8.24	**100**	40.90	22.72	63.63

perfect matches. While the precision presents lower results on Company and Country categories, which are about 58%. The mean value on the sum of perfect and partial matches (TOT) for recall is around 70%, and the mean value on the total matches for precision is equal to 90.65%.

Table 3.6 shows instead the precision and recall comparing our approach with the more traditional clustering algorithms, using the manual concept identification as a baseline. The two measures are calculated for the six categories shown in Table 3.6 (see the legend for details). The results show high performance of our approach which in general overwhelms all the other methods. Let us notice that also the hierarchical clustering presents a fair precision values, they are much lower than our results, in all the categories.

Table 3.6: 500N-KPCrowd-v1.1 dataset - precision and recall computed on BCs w.r.t. hand annotated concepts: comparison between our approach and hierarchical clustering, K-means, PAM

	Clustering	**Categories**					
	Methods (# clust)	**P**	**L**	**O**	**D**	**M**	**Po**
Precision	Our Approach (2089)	98.80	98.40	100	100	96.64	100
	Hier. clust. (2221)	38.11	12.05	41.64	18.11	1	8.82
	K-means (2097)	36.44	12.33	48	20.33	0.66	16.66
	PAM (2143)	47.2	14.4	59.8	27.6	0.8	15
Recall	Our Approach (2089)	88.21	77.00	90.47	100	97.23	100
	Hier. clust.(2221)	38.38	36.54	67.42	57.89	7.20	81.08
	K-means (2097)	19.43	19.78	41.14	34.39	2.54	81.08
	PAM (2143)	27.96	25.66	56.95	51.87	3.38	81.08

Legend: Person (P), Location (L), Organization (O), Date (D), Money (M), Politics (Po).

A similar experiment was also executed on the dataset Reuters-21578. For this dataset, whose size is smaller, the initial parameter FeaTHR = 0 (i.e., the feature space was composed of all the terms from dataset) is a good choice, and ExtrTHR = 5, as stated above. Table 3.7 shows precision and recall computed on BCs by our approach, w.r.t the AlchemyAPI NE, for each category. The results seem interesting: (PeM) are high on many categories; the total value TOT ranges from a minimum value of 50% on category Holiday for recall and 55.73% on category Company for precision, to 100% on the following categories for recall: Continent, Country, Crime, Drug, EntertainmentAward, Facility, HealthCondition, JobTitle, ProfessionalDegree, RadioProgram, RadioStation, Sport, SportingEvent and StateOrCounty; and for precision: City, Continent, Country, Crime, Facility, FieldTerminology, JobTitle, Movie, NaturalDisaster, ProfessionalDegree, RadioProgram, RadioStation, Sport, SportingEvent, StateOrCounty and Technology. The mean value on TOT is 89.37% and more than 70% of the TOT values are greater than 90% for recall. While precision presents a mean value on TOT of 85.92% and 16 categories out of 36 present 100 % as total precision.

Similarly, the test with the manual annotations has been also

Table 3.7: Reuters-21578 dataset: precision and recall on BCs w.r.t. AlchemyAPI NE for different categories

Category	**Precision**			**Recall**		
	PeM	**PaM**	**TOT**	**PeM**	**PaM**	**TOT**
Anatomy	53.65	26.82	80.48	**61.11**	30.56	**91.67**
Anniversary	36.24	**55.1**	**91.34**	26.53	**57.14**	**83.67**
Automobile	21.31	45.90	67.21	**60**	20	80
City	52.08	48.92	**100**	65.96	6.38	72.34
Company	50.82	4.91	55.73	66.67	0	66.67
Continent	**100**	0	**100**	**100**	0	**100**
Country	**100**	0	**100**	**100**	0	**100**
Crime	**100**	0	**100**	33.33	**66.67**	**100**
Degree	94.72	3.12	97.84	7.23	**89.16**	**96.39**
Drug	5.88	72.54	78.43	6.67	**93.33**	100
EntertainmentAward	5	70	75	**100**	0	**100**
Facility	**100**	0	**100**	**100**	0	**100**
FieldTerminology	**100**	0	**100**	43.75	46.87	**90.63**
FinancialMarketIndex	61.57	30.73	**92.3**	50	40.63	**90.63**
GeographicFeature	34.14	36.58	70.73	59.24	27.72	**86.96**
HealthCondition	39.02	31.70	70.73	0	**100**	**100**
Holiday	44.30	20.73	65.04	50	0	50
JobTitle	**100**	0	**100**	**100**	0	**100**
Movie	**100**	0	**100**	58.33	8.33	66.67
NaturalDisaster	25.43	74,57	**100**	0	**91.36**	**91.36**
OperatingSystem	63.41	30.94	**94.35**	61.11	30.56	**91.67**
Organization	22.35	67.07	89.48	26.53	57.14	**83.67**
Person	53.65	26.82	80.48	**60**	20	80
PrintMedia	21.31	45.90	67.21	**65.96**	6.38	72.34
Product	93.63	6.37	**100**	**66.67**	0	66.67
ProfessionalDegree	**100**	0	**100**	**100**	0	**100**
RadioProgram	**100**	0	**100**	**100**	0	**100**
RadioStation	**76.84**	23.16	**100**	33.33	**66.67**	**100**
Region	24.78	67.99	**92.77**	7.23	**89.16**	**96.39**
Sport	**100**	0	**100**	6.67	**93.33**	**100**
SportingEvent	**100**	0	**100**	**100**	0	**100**
StateOrCounty	**100**	0	**100**	**100**	0	**100**
Technology	**93.86**	6.14	**100**	43.75	46.88	**90.63**
TelevisionShow	5	70	75	50	40.63	**90.63**
TelevisionStation	5.88	72.54	78.43	59.24	27.72	**86.96**
Money	34.14	36.58	70.73	0	**91.36**	**91.36**

Table 3.8: Reuters-21578 dataset - precision and recall computed on BCs w.r.t. hand annotated concepts: comparison between our approach and hierarchical clustering, K-means, PAM

	Clustering	Categories					
	Methods	Categories					
	(# clust)	P	L	O	D	M	Po
Precision	Our Approach (2089)	100	100	100	94.83	100	100
	Hier. clust. (2221)	70	68.88	70.74	97.04	74.81	81.11
	K-means (2097)	62	61.33	63	59.66	67.33	72.66
	PAM (2143)	68.76	60.21	62	34.3	67.33	72.66
Recall	Our Approach (2089)	97.54	100	99.01	93.81	99.01	99.09
	Hier. clust. (2221)	92.64	89.85	94.08	77.95	95.06	99.54
	K-means (2097)	91.17	88.88	93.1	48.11	96.01	90.29
	PAM (2143)	85.12	86.95	91.62	27.41	89.35	92.42

Legend: Person (P), Location (L), Organization (O), Date (D), Money (M), Politics (Po).

done on Reuters-21578 dataset and presented in Table 3.8. Our approach outperforms all the other methods, especially on the precision. However, there is a lower gap between our approach results and the other method results, especially if these results are compared to those reported on the 500N-KPCrowd-v1.1 dataset (see Table 3.6).

In order to evaluate the performance of our approach in returning CETs, a further experiment was carried out. AlchemyAPI was used to run on our datasets, enabling the concept extraction option. A list of relevant concepts was returned, ranked by the most relevant to less relevant. Each concept indeed was associated with a relevance value in the range [0, 1]: a value close to 0 means a low relevance, on the contrary, values close to 1 means strong relevance.

To simplify the comparison of CETs with AlchemyAPI concepts, the ranked list was split in some relevance ranges. Each range describes a relevance class. Table 3.9 shows these ranges, skipping the range $[0.1, 0.2]$ (concepts with such low relevance values are not meaningful for the purpose of the experimentation). The relevance ranges have been considered as AlchemyAPI “categories”:

the range (0.8, 1] represents the class of all the concepts that are strongly relevant (i.e., with relevance degree greater than 0.8), because they are the most descriptive of the document collection; the terms whose relevance value is lower than 0.4 belong to the last category defined, the less relevant. In this way, the performance has been still evaluated in term of recall and precision. Precisely, the recall represents the percentage of relevant AlchemyAPI concepts that our system can retrieve, with respect to all the returned AlchemyAPI concepts; the precision, instead, is the percentage of relevant AlchemyAPI concepts that our system can retrieve, with respect to all the concepts retrieved by our system.

Table 3.9 shows the performances of our approach (the rows *Our approach* for *Precision* and *Recall*) on 500N-KPCrowd-v1.1 dataset, compared with the clustering methods. To guarantee a similar parametric setting, the same term matrix (i.e., the same feature set) was given as input to the methods. All methods are optimized properly by analyzing their performance for different number of clusters. Table 3.9 shows also the optimal setting of cluster number (# clust), for each method.

Our approach returns good results especially for the concepts classified in the two medium-high relevance ranges (values between 0.4 and 0.8) where the precision is very high, confirming the effectiveness of our approach in retrieving relevant concepts. These ranges collect the most of the concepts that AlchemyAPI returns and that are enough relevant. Particularly, our approach returns high values of precision as the AlchemyAPI concepts become more relevant (i.e., the concepts with relevance value greater than 0.6 but less than 0.8): it means that our approach retrieves the most of the meaningful AlchemyAPI concepts. Anyway, with concepts that are very relevant in the AlchemyAPI ranking (relevance range 0.8-1.0), the precision tends to be a little lower, even though the value is higher than 60%. Also the recall is quite satisfactory, since in general, more than 40% of relevant concepts have been retrieved. A reason of lower recall might be that AlchemyAPI is based on enhanced IR techniques that use external knowledge bases to return concepts. Thus, it can return additional concepts

Table 3.9: 500N-KPCrowd-v1.1 dataset - precision and recall computed on CETs w.r.t. AlchemyAPI concepts: comparison between our approach and hierarchical clustering, K-means, PAM

	Clustering Methods (# clust)	**Relevance Range**			
		(0.2 - 0.4]	**(0.4 - 0.6]**	**(0.6 - 0.8]**	**(0.8 - 1.0]**
Precision	Our Approach (2089)	6.54	68.20	93.80	61.19
	Hier. clust. (2221)	2.71	30	37.28	26.57
	K-means (2097)	3.47	31.69	41.44	30.17
	PAM (2143)	2.57	30	36.57	26.71
Recall	Our Approach (2089)	34.72	42.69	41.27	41.42
	Hier. clust.(2221)	26.39	36.14	34.54	39.72
	K-means (2097)	31.94	40.36	37.97	42.42
	PAM (2143)	25.00	39.53	36.92	38.88

whose terms cannot be present in the processed text corpus.

The performance of our approach, was furthermore validated, achieving a comparative analysis with the hierarchical clustering. As stated before, the hierarchical clustering is a method of clustering analysis that evidences some similarity with the simplicial complex model. In the agglomerative model, it works as a simplicial complex, linking terms (in general, entities) according to some distance measure. The final structure is a dendrogram that can be interpreted as a concept-based structure and compared with our simplexes structure generated through the iterative execution of Algorithm 1. The same term matrix (i.e., the same feature set) was given as input to both methods, by using the euclidean distance as metric.

As evidenced in Table 3.9, our approach always overwhelms the hierarchical clustering: the recall and precision values, computed for the clustering are lower than our approach on every relevance range; even though the recall returned by our approach is higher on all relevant relevance ranges, the precision is very high, if compared with the results of hierarchical clustering. The reasons of the better results in using the simplicial complex than the hierarchical clustering are mainly ascribed to the way the concepts are identified in the complex structure. The level-by-level construction evidences the strong relations among terms forming structures (simplexes)

Table 3.10: Reuters-21578 dataset - precision and recall computed on CETs w.r.t. AlchemyAPI concepts: comparison between our approach and hierarchical clustering, K-means, PAM

	Clustering Methods (# clust)	**Relevance Range**			
		(0.2 - 0.4]	**(0.4 - 0.6]**	**(0.6 - 0.8]**	**(0.8 - 1.0]**
Precision	Our Approach (461)	7.63	45.20	68.08	53.57
	Hier. clust. (514)	2.66	17.33	27	19.21
	K-means (477)	4.28	25.92	36.07	26.08
	PAM (422)	4.23	21.92	36.54	27.21
Recall	Our Approach (416)	52.17	44.37	50.70	57.78
	Hier. clust. (514)	34.78	34.43	37.67	42.22
	K-means (477)	52.12	44.37	46.98	54.07
	PAM (422)	47.82	42.17	44.19	52.59

that stay unchanged across the levels. This way, basic concepts are clearly identified; moreover, the complex structure maintains basic and extended concepts as well-formed and separate concepts, thus contributing to get an increased number of matchings, when compared with AlchemyAPI concepts.
Table 3.9 shows also a performance comparison with the two well-known partitional clustering: K-means and PAM methods. Let us notice that in some range, K-means returns recall values that are better than the other methods, similar or even better than our approach (range $(0.8, 1.0]$), whereas the hierarchical clustering and PAM have very similar recall value in all the ranges; in general the percentage of relevant concepts that all the clusering methods retrieve, is always around to 40%. This fact confirms that, more likely AlchemyAPI returns additional ad-hoc defined concepts.
Differently, the precision value returned by our approach far exceeds all the other methods, confirming that our approach can better recognize more concepts among the retrieved ones, than the other clustering methods.

Finally, Table 3.10 shows the recall and precision computed on CETs for Reuters-21578 dataset, with respect to the relevance ranges defined for AlchemyAPI concepts. As before, a comparative analysis between our approach and the hierarchical clustering, K-means and PAM is also shown. The performance results confirmed the trend already shown in Table 3.9 on the previous dataset. Specifically on this dataset, the performance of our approach was completely satisfying with *very relevant* concepts (i.e., with relevance value greater than 0.6); the precision values of our approach indeed, far exceed those ones returned with the hierarchical clustering, K-means and PAM methods. On this dataset, k-means and PAM generally showed better recall values than the hierarchical clustering, even though the performance of our approach overwhelmed all the other methods, in the high relevance ranges. In nutshell, the proposed framework shows good performance in extracting accurate (basic and extended) concepts, that are also well characterized by additional terms that describe the context. Differently from AlchemyAPI, it works exclusively on the input corpus, without any additional external supporting tool.

3.3 ...to multimedia data generated by devices

3.3.1 Knowledge acquisition from UAV videos: the problem

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 [70] and event detection [71] 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 [72], [73], but also to achieve optimal trajectory tracking [74] and alleviate vehicle routing problems [75]. Although many tracking algorithms can deal with occlusion, split objects, shadows and reflection, object tracking suffers in object labelling [76]; 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 [77] or crowds [78]; vehicles [79], or their relations [80] and the main applications in this area converge on a single type of scenario [81], [82].

Although scene comprehension from moving camera is a hot topic for improving decision-making processes and supporting video-tracking activities like moving object detection and tracking [70], [71], [83], 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 the object detection and tracking activities, and high-level scene interpretation. 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 [84, 85, 86, 87]. Therefore, consolidated fixed camera techniques, such as the background subtraction [88], 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 [81], [89], [90]; camera movements are constant or constrained; tracking is carried out on only one object [77], [79]. Some studies also achieve object recognition by object classification in predefined classes, even though many issues as low resolution [91], motion blur [92], prohibitive camera shots [93], [94] need to be addressed.

For scene understanding, many works propose pattern recognition methods to recognize scene elements or regions [95, 96]. 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. Furthermore, a camera-equipped UAV can take many different environment types with many different typologies of moving objects doing specific activities. Generally, most of these methods work exclusively on low-level pixel-based data, such as colour, shape and position. For instance, 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 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. Therefore, the information on the trajectory is not enough to understand scene object interactions. Further data to discriminate contexts are required. Furthermore, there is a need of methods capable of processing data at different levels of detail (from raw video data to contextual data) to extract thorough knowledge on the scene and support the situation comprehension.

3.3.2 Research questions

Beyond classic tracking-related issues, such as resolution, camera shaking, illumination change and appearance change of the background [72, 73, 97, 98], scene knowledge extraction from UAV videos can be compromised by the camera movements, causing a lack of reference points to interpret scene object movements. Tracking data can also be not enough to depict scene object interaction and situations coming from them. Then, summarizing, the issues related to the knowledge extraction from UAV videos generate the following questions:

- tracking data are not enough to recognize objects and events. What kind of information is required to improve these tasks ? Then, how to integrate tracking data with the new acquired information ?
- Tracking detects the object trajectory, but how can the trajectory be used to explain interactions among objects ? And between the object and the environment ?
- Tracking detects the mobile objects, but fixed elements in the scene can be useful for object labeling and event detection. How can environmental data be acquired and used to improve these tasks ?
- Machine learning methods require great amounts of training samples and can have bad performances. How can object labeling and event detection tasks be improved ?

All these issues require methods to process data and achieve high level comprehension of the scene. In order to get a deeper knowledge of the environment, this dissertation discusses 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 behaviours happen, nor it is able to quantify the chance of an event occurring [4]. A synergistic approach that exploits consolidated methodologies (for example, deep learning-based methods) could alleviate the issues related to the foreseeing of future unexpected/unpredictable events.

The role of Semantic Web technologies is crucial in the knowledge representation, yielding the knowledge in the form of concepts and relations among them; they encode the vagueness of natural language (embedded in the linguistic terms) by identifying conceptual entities in the resource content. The ontology is a specific artifact designed to represent a real world domain by explicit, well-defined concepts, that presuppose a shared view between several parties [99, 100]. It gathers concepts from the real world by means of unambiguous and concise coding [101, 102]. At the same time, it allows capturing the terminological knowledge that sometimes embeds imprecise information, supporting the management of semantic data and the intrinsic ambiguity in their theoretic representation model, providing enhanced data processing and reasoning, and then supplying a suitable conceptualization that bridges the gap between flexible human understanding and hard machine-processing [103]. The next chapter explores the synergistic use of semantics with tracking for knowledge extraction from UV videos.

Chapter 4

Knowledge extraction from UAV videos

4.1 Overview

As stated in the previous chapter, object labeling and event detection through UAVs suffer from some issues basically related to the lack of reference points to interpret the object trajectories. Tracking data alone is not enough to correctly interpret interactions among scene objects.

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 taken by flying UAVs).

The reminder of this chapter explores Semantic Web technologies application to tracking data for knowledge extraction from video in Section 4.2. Then, in Section 4.3, a knowledge representation model for UAV videos is presented and discussed.

4.2 Semantics applied to tracking for scene knowledge extraction

Since Semantic Web technologies allow the modeling of high-level knowledge, they can be used to enrich tracking raw data with high-level information. This way, Semantic Web technologies can 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 the allowed range) and eventually, to wisely support a decision.

Semantic Web technologies have already been used in combination with video-tracking, but with fixed camera applications. Semantic Web technologies are mainly used for data fusion of low-level data coming from different sensors [113], and for data fusion between low-level data and contextual variables [114]. The main goal of methods proposed in literature is to alleviate tracking problems like occlusion, grouping, shadowing, etc. [114], [115], [116], [117]. They are also used to support object detection proposing semantic segmentation techniques to classify pixel regions in predefined classes [118]. Semantic segmentation often involves deep learning techniques to classify pixels in pre-determined categories. These methods report good performances on detecting environmental features [119], even though they require many pre-acquired training samples [120].

In [114], 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, firstly, 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.

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, roads, in general, points of interest (POIs) to better identify and then label the tracked objects. 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 modelling 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.

The next section discusses a knowledge-based approach to UAV knowledge extraction from video. The approach exploits the synergy between the tracking methods and semantic technologies to bridge the object labelling gap, enhance situation awareness, as well as detect and classify simple alerting events. 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 events.

4.3 A model for knowledge extraction from UAV videos

Figure 4.1 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 [121].

The main input is a video recorded by a flying drone with an on-board installed camera (top left of the figure). 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 the figure): it is composed of specific places called Points of Interest (POIs), which are fixed geo-referred points or areas retrieved with Google Maps service, lying in the area where 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 sub-modules. 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.

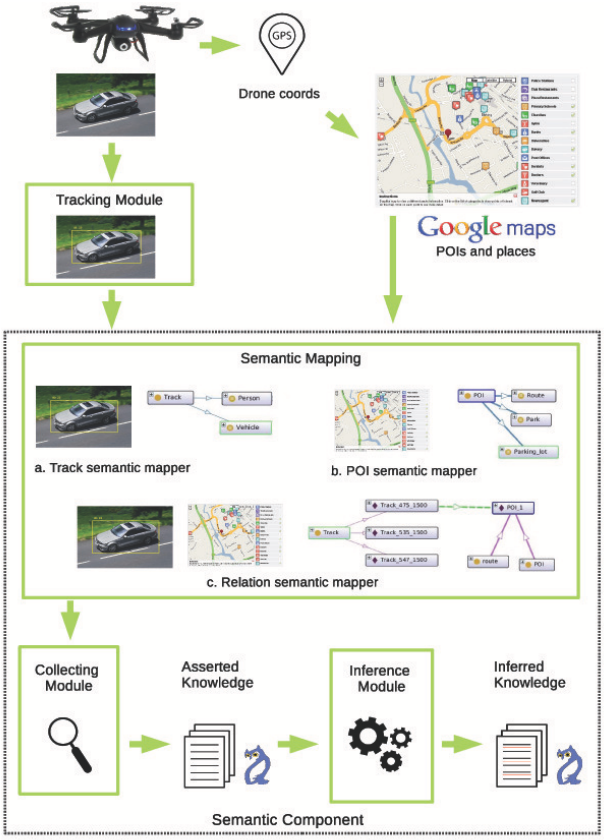
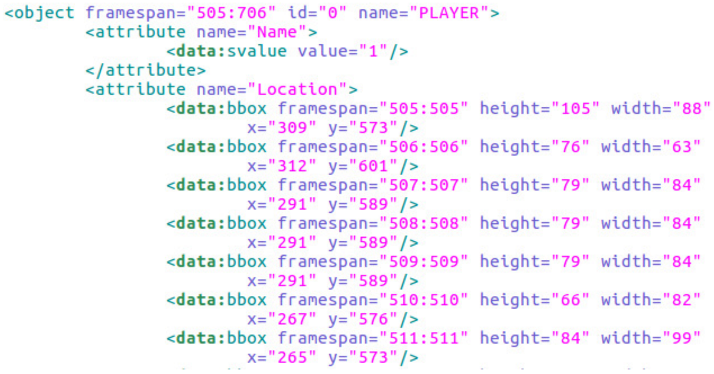


Figure 4.1: A logical overview of the framework

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.



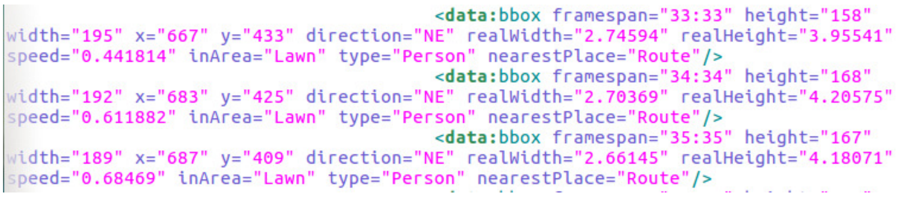
```
<object framespan="505:706" id="0" name="PLAYER">
        <attribute name="Name">
                <data:svalue value="1"/>
        </attribute>
        <attribute name="Location">
                <data:bbox framespan="505:505" height="105" width="88"
                        x="309" y="573"/>
                <data:bbox framespan="506:506" height="76" width="63"
                        x="312" y="601"/>
                <data:bbox framespan="507:507" height="79" width="84"
                        x="291" y="589"/>
                <data:bbox framespan="508:508" height="79" width="84"
                        x="291" y="589"/>
                <data:bbox framespan="509:509" height="79" width="84"
                        x="291" y="589"/>
                <data:bbox framespan="510:510" height="66" width="82"
                        x="267" y="576"/>
                <data:bbox framespan="511:511" height="84" width="99"
                        x="265" y="573"/>
```

Figure 4.2: Bounding boxes of a scene object in the tracking output file.

4.3.1 Tracking, scene object and area classification output

Tracking is used to detect mobile scene objects, such as people, vehicles, animals from the environment. The tracking algorithm detects BLOBS from frames, BLOB stands for Binary Large OBject and refers to a group of connected pixels in a binary image. Therefore, the tracking algorithm generates a bounding box as a rectangle on each BLOB for each frame. The generated bounding boxes in a frame represent mobile entities in the scene. Each bounding box is also provided with an ID number identifying a specific object. Bounding boxes with the same ID in distinct frames identify the same scene object. Therefore, bounding boxes with the same ID in successive frames allow to reconstruct the trajectory of a scene object in the video.

The tracking output is stored in an XML-like file representing the information on the generated bounding boxes with specific tags. The main tags contain information about the video, such as the framerate, length, number of frames. The tags *object* contain data about the bounding boxes generated for each scene object. Figure 4.2 shows an example of these tags in the tracking output file. The tag *object* represents the scene object with a specific ID number. The child tag *attribute* with attribute *Location* contains a series



```
                                        <data:bbox framespan="33:33" height="158"
width="195" x="667" y="433" direction="NE" realWidth="2.74594" realHeight="3.95541"
speed="0.441814" inArea="Lawn" type="Person" nearestPlace="Route"/>
                                        <data:bbox framespan="34:34" height="168"
width="192" x="683" y="425" direction="NE" realWidth="2.70369" realHeight="4.20575"
speed="0.611882" inArea="Lawn" type="Person" nearestPlace="Route"/>
                                        <data:bbox framespan="35:35" height="167"
width="189" x="687" y="409" direction="NE" realWidth="2.66145" realHeight="4.18071"
speed="0.68469" inArea="Lawn" type="Person" nearestPlace="Route"/>
```

Figure 4.3: Object and area classification results added to bounding box tags in the tracking output file.

of tag *data:bbox* representing the bounding boxes of the object in distinct frames. The attribute of the tag *data:bbox* describe the features of a bounding box, as follows:

- The attribute *framespan* represents the number of the frame in which the bounding box is present

- The attributes *height* and *weight* represent the set width and the height of the bounding box, respectively

- The attributes x and y represent the position in pixel of the bounding box in the frame.

The object classification, introduced in [122], is also applied to the tracking data to detect the object identity among people, vehicles or unknown categories. This way, each bounding box of a scene object can be associated with a type expressing the scene object identity. The application of area classifiers, introduced in [122], allow to detect the identity of places where the scene objects move. Then, each bounding box of a scene object can be associated with the area it stays or areas in its surroundings. Classification results are added to the tracking output to enrich the information related to each bounding box. Figure 4.3 shows the new attributes added to each tag *data:bbox*, that relate the classification results to the bounding box. They are reported as follows:

- the attribute *type* associates the bounding box of a scene object with the scene object identity

- the attribute *inArea* associates the bounding box with the type of environment on which it is moving
- the attribute *nearestPlace* associates the bounding box with the type of environment which is the closest in its surroundings.
- the attribute *direction* associates the bounding box with its direction according to its movement from the previous frame
- the attributes *realWidth* and *realHeight* associate the bounding box with real dimensions of the bounding box.

4.3.2 TrackPOI ontology

After the video analysis tasks have been accomplished, the data flow passes to *Semantic component* that generates high-level knowledge on the whole scenario present in the video. Semantic Web technologies are used to code tracking data and higher-level data on the scene environment into semantic statements. To this purpose, the TrackPOI ontology (see Figure 4.4) is used to describe the scenario at a semantic level. The ontology, written in OWL language, models the scene as composed of two main entities, which are the mobile and the fixed objects. The former are modeled as instances of the *TrackPOI:Track* class, while the latter as instances of the *TrackPOI:POI*. These two main classes can have subclasses representing specialized types of mobile and fixed objects. Instances of these classes can be related by properties to build contextual knowledge on the scene by bridging video raw data with higher-level information retrieved from sources external to the UAV. The next sections introduce the instantiation of the two main types of scene objects and the relations among them to build a spatial/temporal context useful to perform object labeling and describe object interactions.

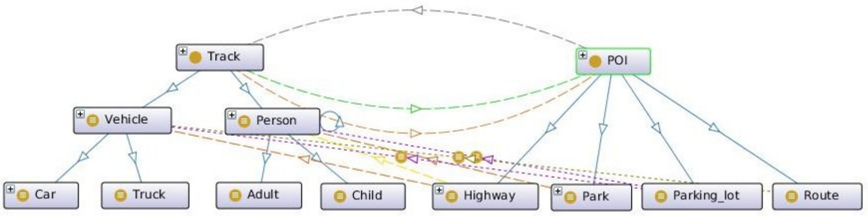


Figure 4.4: The TrackPOI ontology schema

4.3.3 Mobile objects: TrackPOI:Track class

The Track semantic mapper is in charge of converting the tracking output, provided by the Tracking Module, into ontological statements. A tracked scene object is a dynamic object present in the video, that moves in the scene. We define track as the bounding box marking the tracked scene object in a frame. A tracked scene object can rapidly change from a moving to fixed state and viceversa in a few of frames. The mobile objects, that populate the scene, can be of different type. In most cases, the mobile objects move autonomously in the environment, and they often are living beings, such as humans or animals, but they can also be non-living things carried, pushed or driven by living beings like vehicles, shopping carts, etc. Formally, $\hat{M} = \{\hat{o}_1, \hat{o}_2, ..\}$ is the set of the mobile objects moving in the video scene. 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 $j = 1, \ldots, n$ of the video. The tracks of each object $\hat{o}_i$ are, respectively, directly coded into *Track* class instances of the TrackPOI ontology.

The instantiation of the tracks is automatically performed by reading the tracking output. 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 (Figure 4.4), where *Track* is the ontology class modelling the tracked objects. The *Track* individual describes a specific track in the frame, by means of the following properties:

- *TrackPOI:trackID*, bounding box ID
- *TrackPOI:trackName*, track name
- *TrackPOI:width*, *TrackPOI:height*, bounding box dimensions
- *TrackPOI:X*, *TrackPOI:Y*, bounding box position as coordinates of the top left point of the bounding box
- *TrackPOI:width_m*, *TrackPOI:height_m*, track real dimensions (width, height) in meters
- *TrackPOI:hasSpeed*, track speed
- *georss:point*, track position (GPS coordinates)
- *TrackPOI:frame_ID*, ID of the frame which the track is related to

Information about track real dimensions and speed, are calculated by using well-known frame-scene mapping models, such as the Pinhole Model [124, 125].

A track instance is generated for each frame where the track is in. Tracks, describing the same objects in distinct frames, have the same ID value. Therefore, the scene object is identifiable through frames by its track ID. 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, representing track data in the video frames and the calculated ones in the real scene, are added to the knowledge base (see the ontology in Figure 4.4).

Since TrackPOI ontology is designed to deal with various outside environments mainly populated by humans, animals and vehicles, these mobile object types are modeled by the ontology as subclasses of the *Track* class. The *Track* class has three subclasses modeling three specialized types of track: $Vehicle$, $Person$ and $Unknown$. Therefore, if data about the track identity is available, the $Track$ instance is also coded as instance of one of the $Track$ subclasses. As stated in Section 4.3.1, the tracking output file can be also annotated with classification results. Then, if classification recognizes a tracked object as person, the $Track$ instance will be also a $Person$ instance. Otherwise, the object recognized as Vehicle will trigger the generation of a $Vehicle$ class instance. In case the $Track$ instance is not recognized as $Person$ or $Vehicle$, the $Track$ instance is added as $Uknown$ instance to the knowledge base. If there is no object classification result available, object labeling can be evaluated by using contextual relations between tracks and the environment, as it will be described in Section 4.3.5.

4.3.4 Fixed objects: TrackPOI:POI class

The POI semantic mapper processes POIs and places appearing in video scenes. POIs identify specific places and environmental elements. They are permanently 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 mobile camera-taken video. Furthermore, POIs play another important role in modelling knowledge about the video scenes. In fact, POI data can also be 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. Similarly, if the system detects POI data about a highway, it expects to find vehicles running on it.

The fixed objects, or simply POIs, present in a video scene can be formally represented as the elements of the set $F = \{y_1, y_2, ...\}$. The F elements are environmental static features, identifying areas and localities generally present in outside scenarios, such as roads, parks, parking lots, as well as less extended environmental elements (i.e., stores, ATMs, etc.). Each fixed object in the F set can be easily represented as an instance of the *POI* class from TrackPOI ontology, that models all the fixed elements of the scenario.

Data about the fixed objects can be got through several sources. Among them, GPS-based services like Google Maps [123] can be used. These services provide data about the identity and features of the Points of Interest (POIs), and possible places appearing in the scene.

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. In other words, POIs represent structured data about urban and natural sites (i.e., roads, buildings, business activities, national parks, rivers, mountains, etc.) localized with their own GPS coordinates. The POI data are useful to identify the macro area which the drone is flying over. For each retrieved POI, the POI semantic mapper generates a new POI individual, i.e., an instance of the *POI* class (described by *Track-POI* ontology, Figure 4.4), 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, such as 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, park) and their related information in a similar

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

[2] https://developers.google.com/places/

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

Table 4.1: Some Google places

Places		
airport	amusement_park	train_station
art_gallery	bank	subway_station
bus_station	car_repair	stadium
cemetery	church	parking
gas_station	hospital	school
movie_theater	museum	police
night_club	park	ATM

way to Geocoding API. Some of the place types are shown in Table 4.1.

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 Track-POI Ontology (see the ontology in Figure 4.4).

Beyond the GPS-based services, pixel data can be also used to detect fixed object identities. As stated in Section 4.3.1, an area classifier applied to pixel data can support the identification of areas present in the video. Obviously, classifiers need to be trained on all the environment types that the UAV needs to detect. The areas detected by classification results, that are associated to each track in the tracking output file, can be directly coded into POI instances.

4.3.5 Spatio/temporal relations

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 scene elements appearing in the frames by the analysis of the possible interactions between them in the evolving scene. The investigated relations can hold between a track and a POI, and among two tracks, which, respectively, represent two distinct scene objects. As tracks and POIs, relations can be coded as asserted knowledge, which can serve the building of a context useful to depict the UAV monitored scenario. In this dissertation we focus particularly on the positional relations, which make possible to analyse 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 analysed relations cover the main grammar prepositions of place: *front of, behind, near, between, in, 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 analyses 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 five different specializations of the predicate *TrackPOI:hasRelationWith*, that are modelled in the TrackPOI ontology. They are detailed as follows.

- *TrackPOI:isInTheAreaOf*: this predicate is used to relate two entities, x and y. The statement x *TrackPOI:isInTheAreaOf* y (see Figure 4.5 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.

- *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 Figure 4.5 b).

- *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 the track is in, where x is the track and p the cardinal point representing its direction (see Figure 4.5 c).

- *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 details, 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 Figure 4.5 d).

- *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 Figure 4.5 e).

- *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 Figure 4.5 f).

These new specialized relations, such as *TrackPOI:inFrontOf*, *TrackPOI:isBehind*, *TrackPOI:isInBetween*, etc. allow a better characterization of the positional context. The spatial relations are also related to video time. In fact, the spatial relations among the objects are timed according to the video time, relating the track instance to its frame time instant with a specialized property. This property is *TrackPOI:hasTime*, which is associated with each track in the video.

UAV GPS data is not always available, or precise according to the service employed. Therefore, a robust relation asserter needs to use other services to estimate relations among tracks, and especially between a track and a POI. To this purpose, the area classification, introduced in Section 4.3.1, has also been employed to code relations among scene objects into ontological statements. Beyond the GPS-based spatial relations listed in Figure 4.5, further relations are generated according to the area classification results. The classification result-based relations added are two, respectively, expressed by using two properties: *TrackPOI:inArea* and *TrackPOI:nearestPlace*. The *TrackPOI:inArea* property codes the relation between the scene object (Track instance) and the area (POI instance) where the track is moving. Similarly, the POIs or places lying in the track neighbourhood are related to it by using the property *TrackPOI:nearestPlace.* A track and a POI are related with this property if the distance between the object and the place contour lies under a reasonable threshold, which is

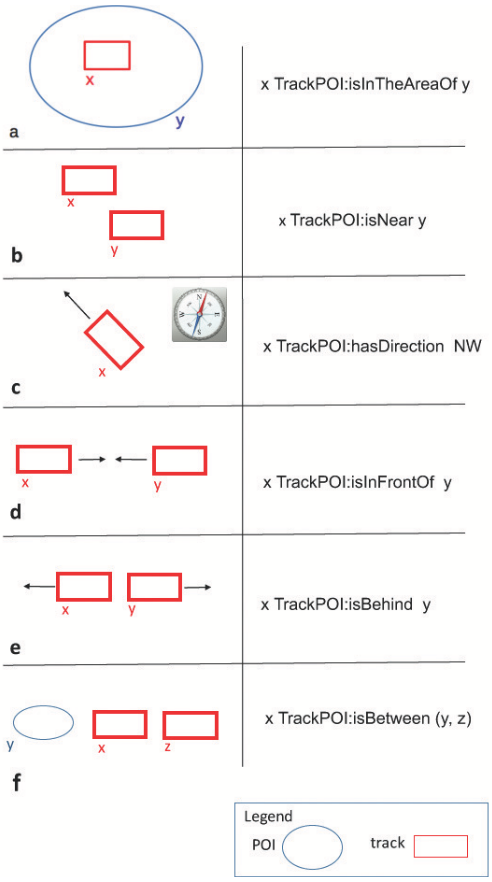


Figure 4.5: Relation Predicates

related to the features of the involved POI. The *TrackPOI:inArea* and *TrackPOI:nearestPlace* properties describe the context of the track movements by relating them with the POIs and the video time. 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.

4.3.6 Collecting and reasoning by using the TrackPOI representation: a case scenario

The Collecting Module completes the assertion process, taking as input the asserted statements produced by the semantic mappers. It synthesizes 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 (Figure 4.4). 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 towards each other. *TrackPOI:inFrontOf* is indeed a symmetric property.

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. Figure 4.6 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. Listing 4.3 describes all the tracks appearing in the frame by ontological statements (black lines 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. Lines 6-18 show statements on the POI entity named POI_1: it is an individual of the *TrackPOI:POI* class (line 8); it is a route (line 9). It is also described by assertions

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

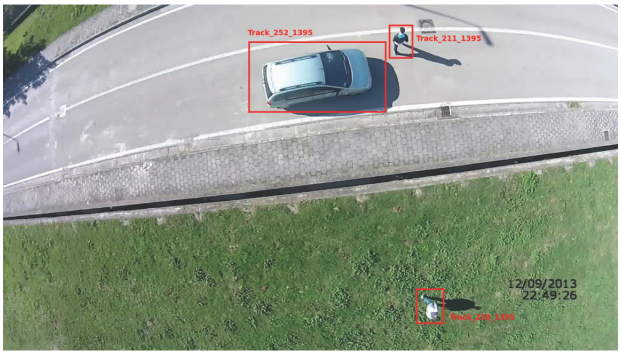


Figure 4.6: Frame 1395 from Video 1

on the area and the geo-position (lines 16, 17), and other Google Maps data like street address, country, administrative area level and postal code (lines 10-15). Tracks are described in black lines 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* (lines 21-22, 39-40) with object dimensions and movements data like width, height and speed (lines 28-30, 46-48) and GPS track position (lines 31, 49). The remaining assertions on track entities are about movements and relations. Specifically, statements in lines 23-27 and 41-45 describe the geometrical or positional relations that the track has with other tracks and POIs in the same frame. Direction statements express the track direction from the previous frame with compass points (lines 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.

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 labelling, 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 SWRL (Semantic Web Rule Language)[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 Figure 4.4). Listing 4.1 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 (lines 4, 6), and a speed which is acceptable for a moving human being (line 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 (lines 2, 3). In order to define the right relations between entities, allowable *Track* individuals for a certain POI have to be specified. Listing 4.2 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 can not 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.

```
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 4.1: Equivalent class restriction for Person class

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

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

Listing 4.2: Equivalent class restriction for Park class

Restriction-based reasoning supports the object labelling, but the expressive power is not suitable for situation understanding. Consequently, rules have been designed to recognize 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:

$$\textit{Person(?x)} \wedge \textit{Route(?y)} \wedge \textit{Vehicle(?z)} \wedge \textit{isInTheAreaOf(?x, ?y)}$$

$$\wedge\ \textit{isInTheAreaOf(?z, ?y)} \wedge \textit{isNear(?x, ?z)} \rightarrow \textit{isInDangerOn(?x,?y)}$$

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 detection of an event that can lead to 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 Figure 4.6, new assertions are generated on the tracks Track_211_1395 and Track_252_1395: Listing 4.3 shows the final augmented knowledge, with the inferred statements in red.

These assertions state that Track_211_1395 is actually a person (line 35): so far the previous asserted statements (lines 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 (line 52) and more specifically a car (line 53). Statements on the situation are also deducted: Track_211_1395 is in an alerting situation, since it is on the route POI_1 (line 36).

```
 1 @prefix trackpoi: <http://www.semanticweb.org/danilo/ontologies/2016/1/ .
 2 @prefix owl: <http://www.w3.org/2002/07/owl#> .
 3 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
 4 @prefix georss: <http://www.georss.org/georss/> .
 5
 6 # POI_1 triples
 7 TrackPOI:POI_1 rdf:type owl:NamedIndividual ,
 8 TrackPOI:POI_1 rdf:type TrackPOI:POI .
 9 TrackPOI:POI_1 rdf:type TrackPOI:Route .
10 TrackPOI:POI_1 TrackPOI:Administrative_area_level_1 "Campania" .
11 TrackPOI:POI_1 TrackPOI:Administrative_area_level_2 "Provincia di Salerno" .
12 TrackPOI:POI_1 TrackPOI:Administrative_area_level_3 "Fisciano" .
13 TrackPOI:POI_1 TrackPOI:Country "Italy" .
14 TrackPOI:POI_1 TrackPOI:Postal_code "84084" .
15 TrackPOI:POI_1 TrackPOI:Street_address "Anello Esterno, 84084 Fisciano SA, Italy" .
16 TrackPOI:POI_1 georss:box "40.775762,14.787031,40.772937,14.785706" .
17 TrackPOI:POI_1 georss:point "40.7743843,14.7860267" .
18 TrackPOI:POI_1 TrackPOI:Name "Anello Esterno" .
19
20 # track_211_1395 triples
21 TrackPOI:Track_211_1395 rdf:type owl:NamedIndividual .
22 TrackPOI:Track_211_1395 rdf:type TrackPOI:Track .
23 TrackPOI:Track_211_1395 TrackPOI:hasRelationWith TrackPOI:POI_1 .
24 TrackPOI:Track_211_1395 TrackPOI:hasRelationWith TrackPOI:Track_239_1395 .
25 TrackPOI:Track_211_1395 TrackPOI:hasRelationWith TrackPOI:Track_252_1395 .
26 TrackPOI:Track_211_1395 TrackPOI:isNear TrackPOI:Track_252_1395 .
27 TrackPOI:Track_211_1395 TrackPOI:isInTheAreaOf TrackPOI:POI_1 .
28 TrackPOI:Track_211_1395 TrackPOI:hasHeight 58.309975411032354 .
29 TrackPOI:Track_211_1395 TrackPOI:hasSpeed 3.2 .
30 TrackPOI:Track_211_1395 TrackPOI:hasWidth 38.51992315031834 .
31 TrackPOI:Track_211_1395 georss:point "40.77454810242632, 14.78530388911672" .
32 TrackPOI:Track_211_1395 TrackPOI:hasDirection TrackPOI:N .
33
34 # inferred triples for track_211_1395
35 TrackPOI:Track_211_1395 rdf:type <https://schema.org/Person> .
36 TrackPOI:Track_211_1395 TrackPOI:isInDangerOn TrackPOI:POI_1 .
37
38 # track_252_1395 triples
39 TrackPOI:Track_252_1395 rdf:type owl:NamedIndividual .
40 TrackPOI:Track_252_1395 rdf:type TrackPOI:Track .
41 TrackPOI:Track_252_1395 TrackPOI:hasRelationWith TrackPOI:POI_1 .
42 TrackPOI:Track_252_1395 TrackPOI:hasRelationWith TrackPOI:Track_211_1395 .
43 TrackPOI:Track_252_1395 TrackPOI:hasRelationWith TrackPOI:Track_239_1395 .
44 TrackPOI:Track_252_1395 TrackPOI:isInTheAreaOf TrackPOI:POI_1 .
45 TrackPOI:Track_252_1395 TrackPOI:isNear TrackPOI:Track_211_1395 .
46 TrackPOI:Track_252_1395 TrackPOI:hasHeight 50.35861512770976 .
47 TrackPOI:Track_252_1395 TrackPOI:hasSpeed 3.4 .
48 TrackPOI:Track_252_1395 TrackPOI:hasWidth 13.428964034055936 .
49 TrackPOI:Track_252_1395 georss:point "40.77412995839289, 14.786379978047478" .
50 TrackPOI:Track_252_1395 TrackPOI:hasDirection TrackPOI:ENE .
51 # inferred triples for track_252_1395
52 TrackPOI:Track_252_1395 rdf:type <https://schema.org/Vehicle> .
53 TrackPOI:Track_252_1395 rdf:type TrackPOI:Car .
```

Listing 4.3: Assertional and inferred knowledge in Turtle statements for POI_1, :track_211_1395 and :track_252_1395. **The inferred triples are in red.**

Chapter 5

UAV comprehension of activities and situations

5.1 Introduction

UAVs are extensively used for research, monitoring and assistance in several fields of application ranging from defense, emergency and disaster management to agriculture, delivery of items, filming and so on. Their performance is often estimated about how accurate and precise is the provided scenario description, ranging from the basic identification of fixed and mobile targets, to recognize target actions that constitute events occurring in the real-time scenario. Especially, when a high-level description of the scenario is strongly desired, UAVs should be able to process the initial tracking data and, by adding environmental information, interpret the scene captured by the on-board camera. Although the human remote control of these vehicles is often decisive to clearly understand the scene and make an action, UAV equipped with such abilities could support human operators in many situations, especially if they are dangerous for humans.

By focusing on a UAV-based surveillance system, video scenario understanding is accomplished gradually through the three hierarchical levels that form the SA [126]: *perception*, *comprehension* and *projection*, starting from data sensing to high-level comprehension.

Figure 5.1 shows a sketched mapping of the three SA levels to our approach of UAV-based surveillance system in a broken car case study. The *perception* level involves all the sensing processes aimed at acquiring data from the video scene (e.g. presence of a car and smoke), the *comprehension* level concerns the high-level understanding about the object interaction in the scenes (e.g. some smoke from car, which probably has broken down), finally the *projection* level involves methodologies to make a decision or to evaluate some possible evolutions of the current scene (e.g., request for roadside assistance).

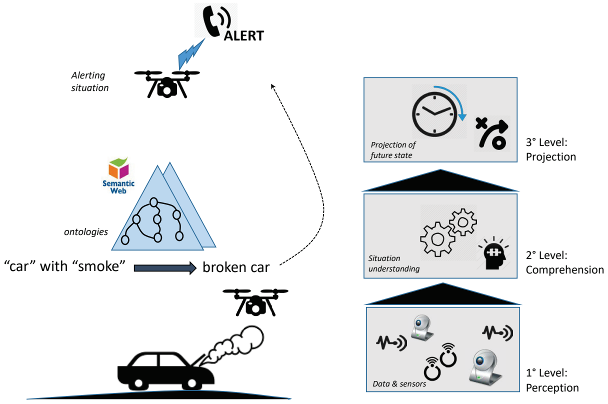


Figure 5.1: The three levels of SA mapped in the UAV-based survillance system

The scenario comprehension requires to analyze low level data and, then, build knowledge on different aspects of the scene, collecting distinct levels of data detail and merge them, increasingly, to get a complete picture of what it is happening [127].

A straightforward interpretation of a scenario requires, as first step, the detection of the main scene actors, such as people, vehicles moving in the scene. Then, the comprehension of their movements and interactions is required to recognize actions or events. Series of events, to which one or more objects participate, depict higher-level activities or situations. This process gradually transforms primitive data (e.g. from sensors or tracking) into high-level information to reach a high-level view of the scenario. Figure 5.2 represents

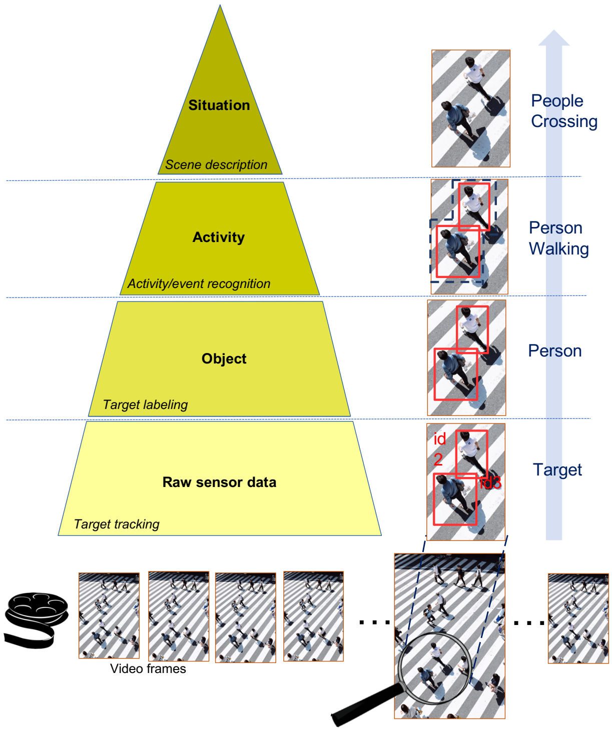


Figure 5.2: Layered knowledge schema on a video scenario

this process. A layered representation describes the incremental knowledge extraction. At the bottom of the figure, the original video frames are processed by tracking algorithms (in the figure, the focus is on a zoomed frame portion), to get target bounding boxes. The *Raw sensor data* label evidences the layer of primitive tracking data acquisition, these data include object dimensions, positions, width and height of bounding boxes, etc., and also possible sensing data if sensors are used to collect data at this layer. Tracked targets are the output of the initial data transformation step. The next layer is defined on the scene object detailed features, obtained through the tracking process or external sources. The *Object* layer is composed of all the recognized targets, including the target identification and classification activities. In Figure 5.2, for example, the targets identified in the video frames are classified and labeled with the name *Person*. In other words, *Person* is the (class) label associated with the bounding boxes identified as *id1, id2*. The *Activity* layer describes the relations between objects appearing in the scene: moving objects can interact with other (moving or fixed) objects, involving actions, movements, or any scene change. For instance, from people movements and interactions it derives that the objects, labeled as *Person*, are *walking*. The upper layer represents the interpretation of the scene at the highest level, through the activities carried out by the named objects in the scene. The layer *Situation* abstracts the object movements in the environment, to achieve a final human-like interpretation of the scene. In this case, the revealed situation is *People Crossing* that is a high-level synthetic description, which is got by condensing the individual activities *Person Walking* of the previous layer. It explains what is happening on the scene, straightforwardly and concisely. This layered knowledge schema can be taken into consideration as a methodological framework to systematically analyse and design systems for video frame scenario interpretation.

The remaining of this chapter is structured as follows: Section 5.2 presents a preliminary extension of the TrackPOI ontology to detect simple events over time by using temporal windows. Section 5.3 goes deeper into UAV activity detection and introduces a framework to detect articulated activities by relating and integrating simpler activities. Finally, Section 5.4 discusses a multi-ontology framework to build knowledge at each detail layer, according to the schema in Figure 5.2, and achieve comprehension of the high-level situations occurring in the scene.

5.2 A TrackPOI extension for event modeling

The TrackPOI ontology, presented in Chapter 4, allows semantic annotation of each frame in the video. Since event detection re-

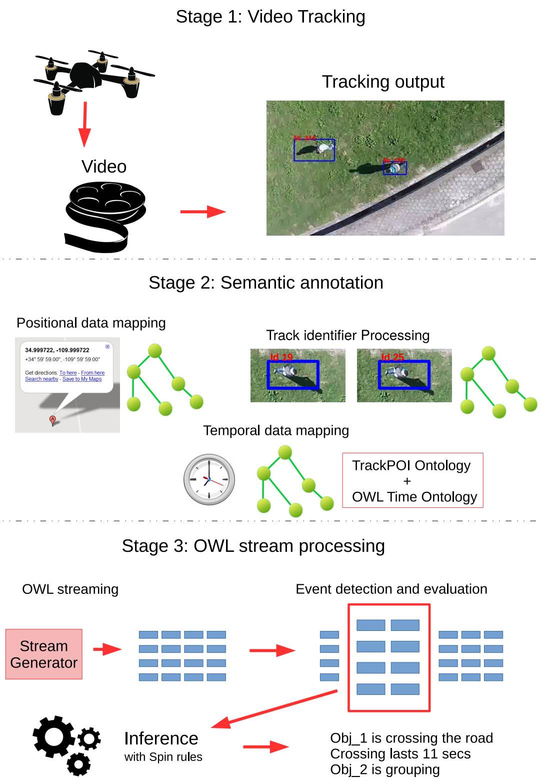


Figure 5.3: A logical overview of the framework

quires temporal evaluations of the object interactions, the ontology model needs to analyse spatio/temporal relations among objects. To this purpose, a preliminary extension of the TrackPOI ontology has been done to allow a temporal modelling of the relations among tracks and POIs and the events. In order to deal with triple processing over time, OWL streaming can be used. OWL streaming allows to process knowledge as a stream of timed triples, that allows the temporal analysis of object relations.

The extension of the knowledge model with the timed analysis of triples is sketched as in Figure 5.3. The figure divides the model into three main stages. The first stage *Video Tracking* codes the input video sequence from the drone on-board camera into a “tagged” video, where object tracks are identified with an ID and framed by a rectangle, namely the *bounding box*, described by its position and size. The second stage *Tracking Data Annotation* is in charge of the semantic enrichment of the video sequence. Then, ontological assertions, generated from the tagged video, feed the knowledge base associated with the video sequence. In details, along with the tracking data, further data related to the geographical position of mobile and fixed objects in the video are retrieved and converted into semantic assertions. These steps can be accomplished by using the TrackPOI representation presented in Chapter 4. Then, besides the coding of the spatial relations, the object tracks are related to each others, with respect to the time. This stage also deals with the ID management, in order to correctly identify an object within the scene when its ID changes due to camouflage or sudden disappearance. The last stage *OWL Stream Processing* aims at processing the collected knowledge base in the form of a stream of ontological statements processed by a temporal window-based application. The temporal window selects sets of consecutive ontological triples according to their temporal ordering to detect relevant spatial/temporal events occurring in the video. Further details about each stage are illustrated in the following sections.

5.2.1 TrackPOI event temporal modeling

According to the TrackPOI ontology, discussed in the previous chapter, the output of tracking, along with target classification data, is coded into semantic assertions. The TrackPOI ontology has been designed to describe dynamic scenarios, where mobile and fixed objects move and interact with each other. The mobile objects in the ontology can be people, vehicles, animals or things moved by the people, which are detected by applying tracking algorithms. The *Track* class represents the bounding box marking the detected object (viz., the track) in each frame of the video. Therefore, each detected object in a frame sequence is represented as a series of *Track* class instances with the same ID value. *Track*

is a general class of the TrackPOI ontology and includes all the recognized moving objects. It needs to be specialized in order to identify instances of its own subclasses, such as *Person* and *Vehicle*. Thus, according to classification results, a *Track* instance can also be a *Person*, *Vehicle* or *Unknown* instance.
The fixed scene objects include environmental features, such as rivers, buildings, stores, etc. The fixed objects are coded as Points of Interest (POIs) retrieved by Google Maps service. In the figure, some fixed objects, namely *Highway*, *Route*, *Park*, *Parking_ lot* are represented as the sub-classes of the *POI* class. TrackPOI uses *GeoRSS* ontology to model POI GPS data and also employs *Time* ontology to represent the instant of a track instance.
TrackPOI defines also the spatio/temporal relations among tracks and POIs in a video scene. Relation modelling allows to describe the interactions among tracks, and the track movements in the environment. According to the layered knowledge scheme of Figure 5.2, TrackPOI models the knowledge on the lowest layer, dedicated to the mobile objects of the scene. It is in charge of generating assertions on tracking and classification data to describe targets and the elementary movements involving them.

In order to model objects and events with respect to time, TrackPOI has been further extended with OWL Time Ontology[1]. OWL Time ontology defines the class *TemporalEntity* to represent a temporal event composed of succeeding instants. Event duration, starting and ending instants are expressed by the properties *hasDuration*, *hasBeginning*, *hasEnd*. The novel integrated ontology is shown in Figure 5.4.

Recalling the sketch in Figure 5.3, when the positional data mapping, in the stage 2, is accomplished, each moving object is described as a series of bounding boxes, each one related to a specific frame. Then, the temporal data mapping module employs the described ontology schema to model collections of timed tracks (bounding boxes). Each track happens in a precise time instant (τ) corresponding to a specific frame of the video. Therefore, each track instance is provided with the time instant of the frame in

[1] https://www.w3.org/TR/owl-time/

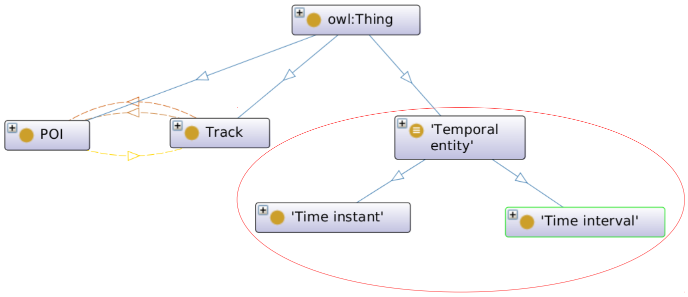


Figure 5.4: The TrackPOI ontology extension with the OWL Time ontology. The TemporalEntity class allows to model the events.

which it appears, then, all the bounding boxes with the same ID and related to the same object are collected. The obtained track collection represents the object temporal evolution in the scene. All the positional relations which link the same couple of tracks or track and POIs through succeeding time intervals are provided with a time duration. Time duration is expressed with a couple of starting and ending time instants. Just to give an example, let us consider the following extracts, describing spatial and temporal aspects between a track and a POI in form of triples:

Tr_1_36 isNear POI_1.
Tr_1_36 hasTime τ_{23}.
Tr_1_37 isNear POI_1.
Tr_1_37 hasTime τ_{24}.
Tr_1_38 isNear POI_1.
Tr_1_38 hasTime τ_{25}.

Precisely, the object with ID equal to 1 (track instances starting with Tr_1) is near POI_1 in the frames numbered 36, 37 and 38, at the time instants τ_{23}, τ_{24} and τ_{25}, respectively. The relation *isNear* between Tr_1 and POI_1 lasts three time instants, starting at τ_{23} and ending at τ_{25}. Duration time for each relation is calculated with a SPIN[2] rule.

[2] http://spinrdf.org/

SPIN[3] (SPARQL Inference Notation) is a SPARQL based rule language. It defines rules exploiting SPARQL query format which can be used to assert new facts, create new individuals or compute the probability of a certain event (abductive reasoning). The great advantage of using SPIN stays also in the possibility to use, in addition to SPARQL, other languages for defining a rule like JAVA.

Once each relation is provided with a temporal duration, the temporal mapping module adds specific relations defined in OWL Time Ontology, which reinforces relations between temporal events according to Allen's algebra. This algebra is included in the ontology in order to model relations on intervals (e.g., meets, overlaps) for representing qualitative temporal information. Specifically, OWL Time Ontology is used to represent temporal relations between positional statements, modeled in TrackPOI Ontology via OWL properties such as *isNear*, *isInTheAreaOf*, etc. In other words, the temporal relations are meant to describe if two positional relations, as temporal events occurring between time periods, are overlapping, meeting, equal, etc. The temporal relations between the positional statements are calculated by a Spin rule. A rule to check if two events overlap and create instances of two new properties *intervalOverlaps* (and its inverse *intervalOverlappedBy*) in the knowledge base is reported in Listing 5.1.

```
INSERT {
    ?A time:intervalOverlaps ?B .
    ?B time:intervalOverlappedBy ?A .
}
WHERE {
     ?A a time:TemporalEntity .
     ?B a time:TemporalEntity .
     ?A time:hasBeginning ?AStart .
     ?A time:hasEnd ?AEnd .
     ?B time:hasBeginning ?BStart .
     ?B time:hasEnd ?BEnd .
     ?AStart time:inXSDTime ?AStartTime .
     ?AEnd time:inXSDTime ?AEndTime .
     ?BStart time:inXSDTime ?BStartTime .
     ?BEnd time:inXSDTime ?BEndTime .
     FILTER( ?AStartTime<?BStartTime ) .
     FILTER( ?AEndTime>?BStartTime ) .
};
```

[3] http://www.w3.org/Submission/spin-overview/

Listing 5.1: Overlapping events: the query generates triples stating the any two events are overlapping

Lines 2 and 3 under the *INSERT* clause generate, in the knowledge base, the overlap relation between all the possible combinations of existing events belonging to the class *TemporalEntity* (lines 6-7) that have overlapping starting and ending times. In particular, lines 8-11 evidence for two events, among the retrieved ones from the knowledge base, the starting times (variables `?AStart` and `?BStart)` and the ending times (variables `?AEnd` and `?BEnd`) using the properties *hasBeginning* and *hasEnd*, respectively. Lines 16-17 simply filter out all the events that do not overlap.

5.2.2 Event detection by using temporal windows

Triples produced in the *Semantic Annotation* stage feed the knowledge base that in turn is processed by the Stream Generator module, in the *OWL stream processing* stage (Figure 5.3). The Stream Generator is a server that reads all the triples produced so far and selects them according to their temporal ordering, in order to send these triples as a stream of information to a temporal window based application. Each stream packet contains a set of triples happening in the same time interval. The time interval has been set by the Stream Generator to a fixed time amount in seconds, then, before starting the sending process, the server has divided the video time length in a succession of time intervals. The server inserts the triples into packets according to the triple time ordering, and periodically send them to the temporal window.

This stream generator processes information in real time and filters out all the triples representing useless or redundant information, in order to simplify and speed up the inference processing. This stream model is also useful to build real time applications for processing live video streaming.

To discover events throughout the video, a temporal window gradually analyses OWL triples occurring in a set of consecutive time intervals. The window gradually moves from the start to the end of the video evaluating the temporal relations between the events in real time. Precisely, the temporal window application sets the window size in seconds or in terms of the succeeding time intervals to cover. Then, the process starts and the application takes as input the OWL streaming generated by the Stream Generator. As new stream data is received, the window moves by a fixed step forward to analyse new triples. The temporal window is implemented by a SPIN rule which takes the window size, as argument, to choose how many time intervals the window has to process at once. At each shift of the temporal window, the set of triples in the current window are passed to an inference engine, which applies some rules in order to temporally relate the spatial relations among the moving objects and POIs and detect some possible events. These rules include some general spatio/temporal events, such as object grouping and dispersing or crossing, resting in an area for a specific amount of time.

In order to show how the proposed framework recognizes spatial/temporal events, we have produced a video set with some events involving cars and people. The videos are recorded with a DJI F-450 drone, equipped with a full HD resolution camera. The environment captured by those videos is a main road located inside our Campus with some POIs in the surroundings (e.g. department buildings, laboratories, bar, etc.). One of the analyzed videos[4] lasts 165 seconds and shows some students walking on the lawn and near the road, then one of the students crosses the road while some cars are running down the road. The two main spatio/temporal events that occur in this video are: People walking in group (close one to each other) or alone, and a man crossing the road. As stated, the framework codes the spatial relations in temporal events, assigning to each positional relation its duration as a time interval. Then, the formed temporal ordered triples are generated as an OWL streaming by the Stream Generator. The stream of triples is analysed in real time by a fixed size temporal window, which

[4] https://goo.gl/eA95gq

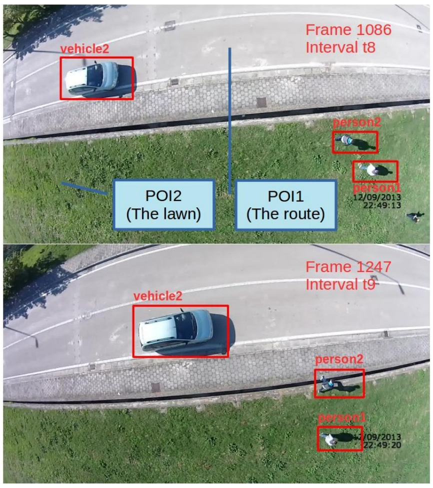


Figure 5.5: Frames 1086 and 1247 from time intervals t8 and t9.

gradually analyses the succeeding time intervals. The temporal window has been built by a SPIN rule. In this example, a temporal interval is fixed equal to 5 seconds; the temporal window is fixed to 20 seconds, then it analyses 4 intervals at a time.

The first event concerns with people grouping and dispersing phenomenons while they are walking. Understanding why they group or divide is a general interesting problem. In Figure 5.5, two students are walking close to each other in the same direction in time intervals between t5 and t8, then they departed keep going alone from interval t9. Our framework detects this event using a SPIN rule which exploits the spatio/temporal knowledge base, collected in the *Semantic annotation* stage. The rule checks if, and for how much time, the two students are walking as group (or close to each other within a minimum distance) or alone. To this purpose, the rule verifies the students' direction, checking if the relation *isNear* between them is lost. If so, a different event is triggered, in that case they are dispersing, otherwise they maintain the relation, staying grouped. The spin rule is presented in Listing 5.2.

```
Select DISTINCT ?event1 ?event2 ?event1After ?event2After
Where{
```

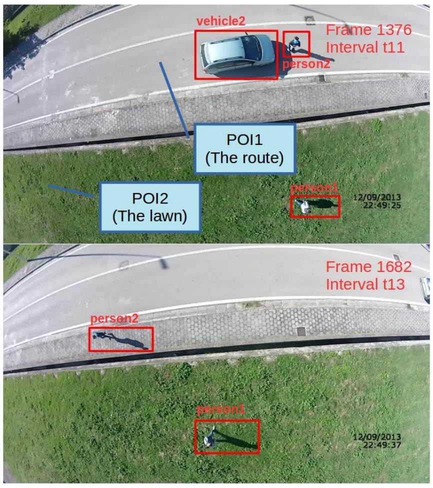
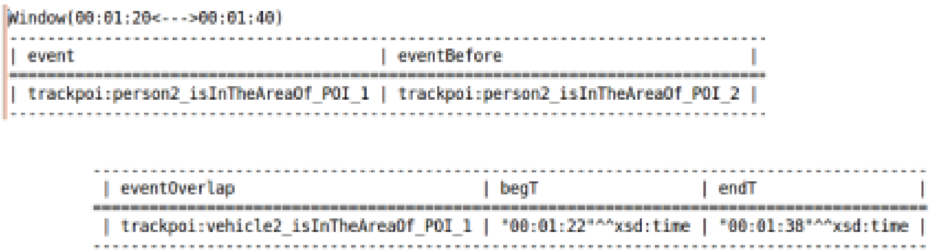


Figure 5.6: Frames 1376 and 1682 from time intervals t11 and t13.



```
Window(00:01:20<--->00:01:40)
-----------------------------------------------------------------------------------------
| event                                  | eventBefore                                  |
=========================================================================================
| trackpoi:person2_isInTheAreaOf_POI_1 | trackpoi:person2_isInTheAreaOf_POI_2 |
-----------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------
| eventOverlap                          | begT                     | endT                     |
=====================================================================================================
| trackpoi:vehicle2_isInTheAreaOf_POI_1 | "00:01:22"^^xsd:time | "00:01:38"^^xsd:time |
-----------------------------------------------------------------------------------------------------
```

Figure 5.7: Crossing event Spin rule result

```
?event1 a time:TemporalEntity .
?event2 a time:TemporalEntity .
?event1 trackpoi:eventOf ?obj1 .
?event2 trackpoi:eventOf ?obj2 .
?event1 trackpoi:isNear ?obj2 .
?event1 trackpoi:isInTheAreaOf ?poi .
?event1 trackpoi:hasDirection ?direction .
?event1 time:intervalMeets ?event1After.
?event2 trackpoi:isNear ?obj1 .
?event2 trackpoi:isInTheAreaOf ?poi .
?event2 trackpoi:hasDirection ?direction .
?event2 time:intervalMeets ?event2After.
?event1After trackpoi:eventOf ?obj1 .
?event2After trackpoi:eventOf ?obj2 .
?event1After trackpoi:hasDirection ?directionAfter .

{MINUS {?event1After trackpoi:isNear ?obj2 .} }
```

```
UNION
{MINUS {?event2After trackpoi:isNear ?obj1 .} }
UNION
{MINUS {?event2After trackpoi:hasDirection ?directionAfter .} }

}
```

Listing 5.2: Spin rule for group event

The second event aims at detecting if a moving object is crossing an area, and how much time he/she spends in doing it. Specifically, in this example the video shows one of the two students crossing the road while a car is arriving. In details, the student walks on the lawn till interval t10, then from interval t11 he starts crossing, finally goes back on the lawn in interval t13.

A SPIN rule has been designed to check if some object track (the student) is crossing the road and how long this event lasts. The rule checks if in some succeeding time intervals there is a $isNear$ relation between the student and the road, followed by the $isInTheAreaOf$ relation and finally a $isNear$ relation again.

The result of the query (in form of table) is given in Figure 5.7. Let us notice from the resulting table that the framework has found a crossing event, involving a moving object ($trackpoi : person2$). The crossing event is composed of two consecutive sub-events: the presence of $trackpoi : person2$ in the area of the lawn ($eventBefore$), followed by the presence of $trackpoi : person2$ in the area of the route ($event$). The duration of the event regarding the presence of $trackpoi : person2$ on the route is calculated according to its starting and ending instants ($begT$ and $endT$). Then, $trackpoi : person2$ crosses the road in 16 seconds. Furthermore, an overlapping event is also detected: a car arriving while $trackpoi : person2$ is crossing ($eventOverlap$).

The generated triples, although good to explain the occurred events, feed the knowledge base with a high number of triples. Consequently, further methods are needed to collect and select information to make reasoning more efficient. Moreover, the detection of situations over time require refined methods to process and fuse knowledge on the UAV detected events.

5.3 Detection of activities

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, this section provides a review of recent literature on the intelligent systems and the use of high-level knowledge to support activity detection.

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 [128]. 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 [120, 129], 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 [123]. Some solutions proposed in literature focus on data fusion, collecting information from heterogeneous sources [130]. 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 [3]. Some solutions are designed for specific environments, so that they present ad-hoc scene ontologies [2], generally exploiting scene segmentation, to represent environment areas and allow deduction of events and object activities [131]. 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 [132] and temporal information [133] to describe the events occurred in the scenario. Some approaches [134], [132] specialise ontologies and query to model places at different levels of granularity (i.e. states, regions, cities) to detect place areas. The approach presented in this section, 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, in this chapter, we discuss a framework that codes and generates high-level knowledge and, then, return a refined set of people or vehicle actions detailing what happened in the observed UAV video.

Recent trends are aimed at building scene ontologies to elicit knowledge about events and activities carried out by the scene objects [131]. 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 [135]. UAVs could fly over different kinds of environments and catch different kinds of objects and situations. Therefore, a priori knowledge [136], or pose classification [137], 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. The framework, proposed in this chapter, defines a general

model for the detection of people and vehicle activities in different contexts.
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 [138, 139]. 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 [140]. The framework, presented in this chapter, instead, firstly defines spatio/temporal relations among the scene objects, and between the scene objects and the environment. Then, it contextualizes these relations by generating knowledge on objects to detect simple activities. Higher-level activities can be then elicited by composing the detected simple activities.

5.3.1 A framework for high-level activity detection

Unlike the most trends in literature, aimed at directly detecting activities by exploiting patterns, the framework, presented in this section, 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.

The Figure 5.8 shows a logical overview of this activity modelling, added as an extra layer to the TrackPOI ontology-based scene representation, that has been introduced in Chapter 4. Specifically, the figure evidences two macro areas: *Scene/Object Video Analysis*, and *Semantic Annotation and Reasoning*. These are the main components of the activity detection 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*

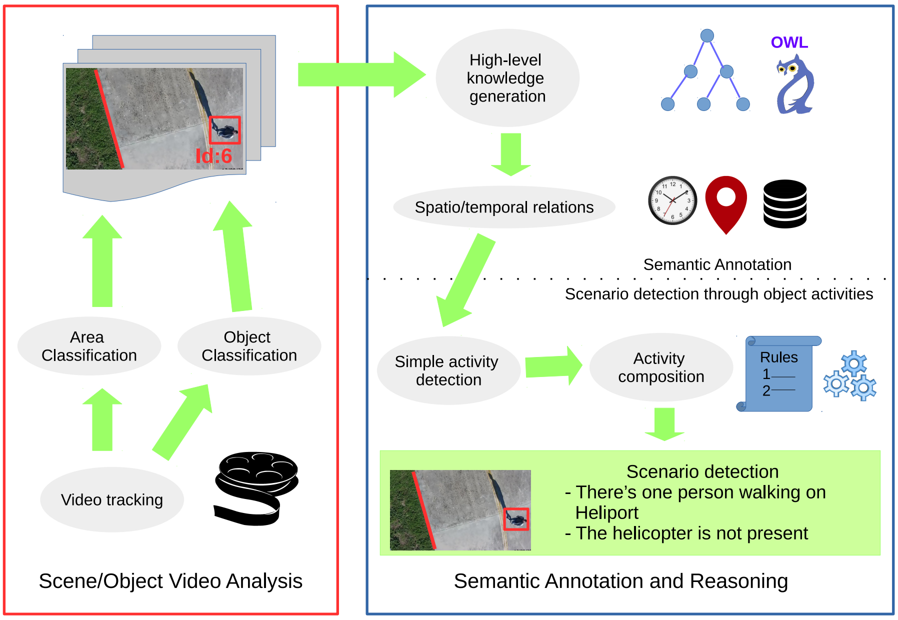

Id:6
Area Classification
Object Classification
Video tracking
Scene/Object Video Analysis
High-level knowledge generation
OWL
Spatio/temporal relations
Semantic Annotation
Scenario detection through object activities
Simple activity detection
Activity composition
Rules
1
2
Scenario detection
- There's one person walking on Heliport
- The helicopter is not present
Semantic Annotation and Reasoning

Figure 5.8: TrackPOI scene ontology extended to complex activity detection

Video Analysis component accomplishes the tracking algorithm on the video to object detection and recognition through frames. As shown in Figure 5.8, 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 tracking and classification output are included in an XML-based file as explained in Chapter 4, Section 4.3.1.

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 the TrackPOI ontology representation, presented in Chapter 4, to semantically describe scenarios populated by moving and fixed scene objects along with their spatio/temporal relations. 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.

The activity detection process starts after the knowledge base has been populated with tracks and POIs, and the spatio/temporal relations have been determined. Therefore, 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 5.9 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 5.9 shows a growing level of semantics, starting from the simple spatio/temporal relations to get a human-like description of complex activities; at

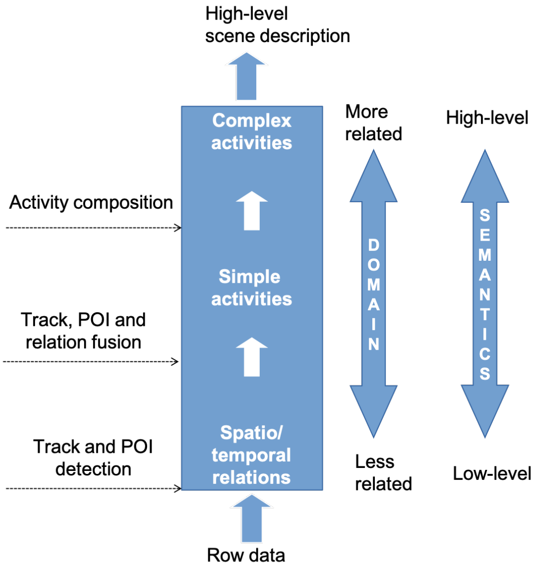


Figure 5.9: Activity modeling: from the row data to the high level scene description, by an incremental definition of the activities

the same time, the domain-dependence increases: as the activities becoming complex, so they are more specialized.

5.3.2 Simple or instant activities

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, respectively, given in Chapter 4, Section 4.3.3 and Section 4.3.4, the simple activity is defined as follows:

Definition 10. ***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*:*

$$S_t = \ < R, \hat{o}_i^t, z >_t \tag{5.1}$$

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

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 some place, 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[5]) rule, shown in Listing 5.3, 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] https://www.topquadrant.com/technology/sparql-rules-spin/

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 CONSTRUCT {
 2     ?this trackpoi:running ?poi .
 3 }
 4 WHERE {
 5     ?this a trackpoi:Vehicle .
 6     ?this trackpoi:track_ID ?id .
 7     ?poi a trackpoi:Route .
 8     ?this trackpoi:inArea ?poi .
 9     ?this trackpoi:speed ?s .
10     FILTER (?s > 0) .
11 }
```

Listing 5.3: Running cars: the SPIN rule detects the simple activity *running* as triples stating that cars are running on a road

5.3.3 Complex activity detection through activity composition

After the simple activities have been 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 5.8) 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 10), the complex activity is a collection of these simple activities/binary relations that hold in a time interval T, more formally:

Definition 11. ***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*:*

$$< C_T, \hat{o}_i, z >_T \;=\; S_{t_1} \wedge S_{t_2} \wedge ... \wedge S_{t_n} \tag{5.2}$$

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

.

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 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 with the same ID that are related to the same object. The collected tracks, each one related to a time instant, represent the object movement. As stated in Section 5.4.2, the scene object, moving in the scene, is

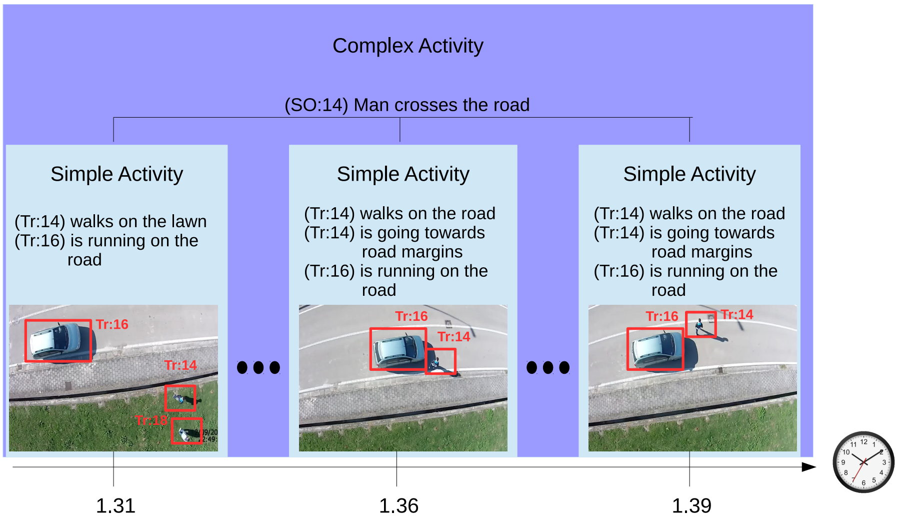

Complex Activity
(SO:14) Man crosses the road
Simple Activity
(Tr:14) walks on the lawn
(Tr:16) is running on the road
Simple Activity
(Tr:14) walks on the road
(Tr:14) is going towards road margins
(Tr:16) is running on the road
Simple Activity
(Tr:14) walks on the road
(Tr:14) is going towards road margins
(Tr:16) is running on the road
Tr:16
Tr:14
Tr:18
Tr:16
Tr:14
Tr:16
Tr:14
1.31
1.36
1.39

Figure 5.10: Activity composition: the complex activity *man crosses the road* is from three simple activities happening over the time $t \in [1.31, 1.39]$

labeled with the name SO, representing the 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, the times associated to the first and last tracks of the SO represent the 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 through its tracks. Then, track simple activities are combined with respect to time and space, to identify the complex activity.
Figure 5.10 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 represent the SO , for example, the tracks identified by *Tr:14* over time (i.e. tracks *Tr:14* in 1.31, 1.36 and 1.39 instants), delineate the moving object *SO:14*. The different consecutive activities, carried out by the tracks compounding the SO, are collected; for example, the *SO:14* is composed of all the activities carried out 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.

The composition model presented emulates an abstraction process which is typical of humans to understand activities and events. Notwithstanding this, the proposed model and a human may have different ways to understand and describe a dynamic scene. In fact, the abstraction process in the model proposed firstly detects mobile objects through video tracking, then it derives simple events by integrating the tracking output with contextual knowledge, and finally complex activities are detected by composing the simple ones with respect to time, space and context. Contrary to the model proposed, humans may have different ways to recognize scene objects (i.e., people, vehicles), detect events, activities and situations from a scene, that may be very different from the abstraction process followed by the model proposed. Furthermore, humans may also describe the scene in different ways, for instance, a person can simply describe the scene in Figure 5.10 by saying "there are a vehicle and man on a road", someone else may describe the same scene at a higher level of abstraction by stating that: "a guy almost got it by a car".

Given these differences in scene understanding, it would be interesting to assess the extent to which our model reflects the actual human behaviours, in terms of object detection, action description and complete scenario comprehension. Therefore, to evaluate how much the model behaviour on activity understanding can be considered correspondent to one of a human, the model behaviour should be compared to real human behaviours. Up to now, the composition model has been tested on expert's annotation in the next section. Anyway, to better assess how much the model behaviour is similar to one of a human, future evaluations will require to compare the model evaluations with those provided by different types of people with various features (age, experience, intelligence quotient, etc) in various environmental contexts.

5.3.4 A demonstrative 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 [123], 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 people and vehicles. Figures 5.11 and 5.12 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 5.8, 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.

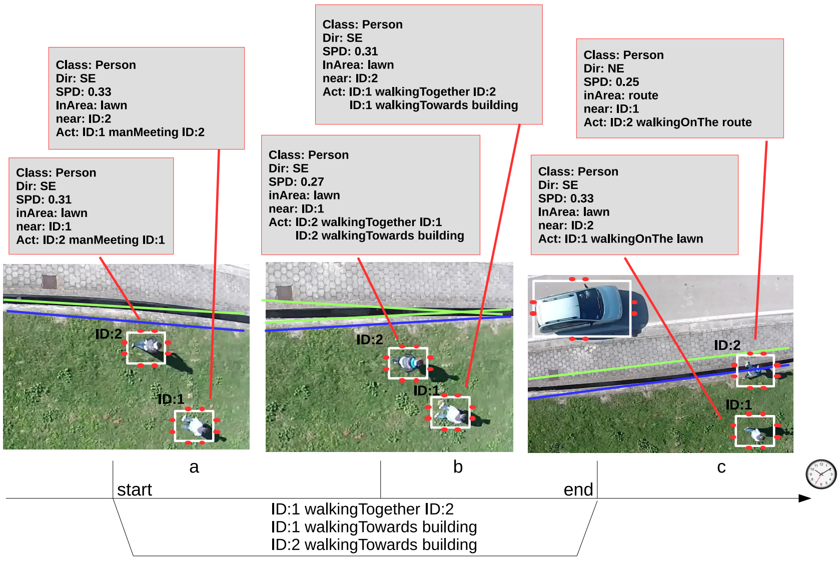


Figure 5.11: System at work: the *walkingTowards* and *walking-Together* activities are detected. The functioning of the system is shown on a video scene showing two people walking together towards a place. The object annotations show the detected relations and activities.

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 5.11), a *near* relation between

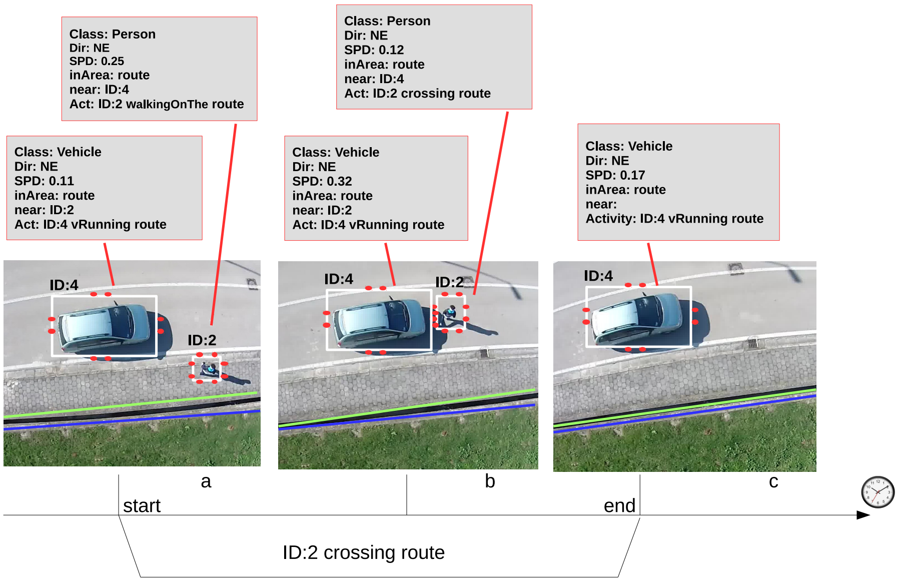


Figure 5.12: System at work: a crossing activity is detected. The functioning of the system is shown on a video scene showing a man crossing the road. The object annotations show the detected relations and activities.

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 5.11a). 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 5.12a).
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 com-

position 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 5.11b). These complex activities start when the *manMeeting* relation is found (Figure 5.11a), 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 Figure 5.11c).

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 5.12b). This activity composition is triggered by the detection of *walkingOnThe* simple activity (*ID:2 walkingOnThe route*), as shown in Figure 5.12a. 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 5.12c 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).

5.3.5 Experimental model evaluation

An experimental evaluation of the activity detection model, proposed in this section, has been conducted and detailed on the following. The approach has been tested 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 achieves good results, evidencing that the synergy between low-level tracking algorithms and high-level semantic scene description leads to performance improvement of the overall system.

The datasets employed for tests are composed of both videos recorded in our campus and downloaded from the Web[6]. They show scenes from several distinct outdoor environments, such as roads, heliports, parks, etc. Also the UAV123 dataset[7] 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 5.1 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 5.1) along with a cumulative number of occurrences, given in the parenthesis. Detailed descriptions about the activities are reported in the Table 5.2 and Table 5.3.

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 5.2 and 5.3 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

[6] https://drive.google.com/open?id=0B75yuWMeqbP5NVloZEIzc05jeW8

[7] https://ivul.kaust.edu.sa/Pages/Dataset-UAV123.aspx

Table 5.1: 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)

Table 5.2: 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 5.3: 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 an area until a certain moment	walkingOnThe, walkingAround
getsInTheCar		Man gets in the car	walkingAround, walkingNear
getsOutOfTheCar		Man gets out of the car	walkingAround, walkingNear

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 5.2 and Table 5.3), 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) [8]. 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 5.13 shows a succession of activities, namely the system-detected (S) and ground truth (GT) activities, placed on the video timeline. Precisely, Figure 5.13a 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 5.13a), (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 (FPs) and false negatives (FNs) in precision and recall computation. As Figure 5.13b 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

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

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 5.13b, activities named VR_1 and VR_a, respectively), then S represents a true positive.

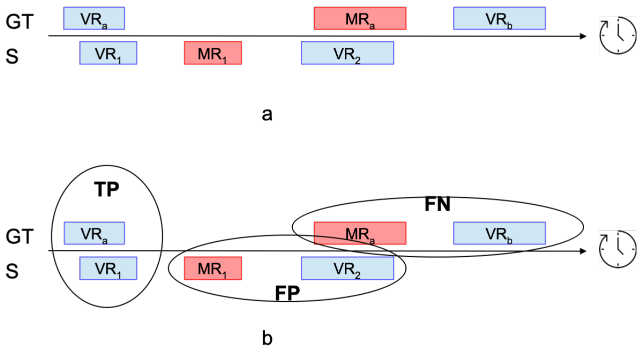


Figure 5.13: An example of activity comparison on vRunnnig (VR) and manRunning (MR): (a) temporal relations between the ground truth and detected activities, (b) True positive, false positive and false negative definition

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) [141]: 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 5.14, JC calculates the ratio in seconds between the

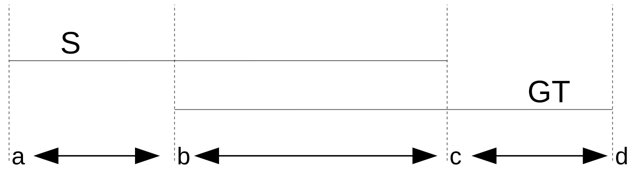


Figure 5.14: Temporal relations between a detected activity (S) and a temporally overlapping ground truth activity (GT) of the same type

overlapping time among the two activities $(c - b)$ and the overall time covered by the two activities $(d - a)$, defined as follows:

$$JC(S, GT) = \frac{c - b}{d - a} \tag{5.3}$$

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 [141, 142]. 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 [143]: 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 5.14 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:

$$TP_S = \frac{c-b}{c-a} \tag{5.4}$$

$$FP_S = 1 - TP_S \tag{5.5}$$

$$FN_{GT} = 1 - \frac{c-b}{d-b} \tag{5.6}$$

Adding up all the TP_S values so calculated, for each detected activity S which overlaps with the ground truth activities GT of the same type, the final TPs are 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 5.4 shows the precision and 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:

Table 5.4: 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

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.

5.4 Multi-ontology design pattern for situation awareness

Recent literature [123, 144] focuses on enhancing UAVs as knowledge-based systems to become aware of situations occurring in a real-world scenario. Knowledge-based methods have been used to perform sensor fusion to integrate heterogeneous data and support various applications [3, 130], such as UAV-driven object detection in video scenes [140, 145]. Cognitive models have been proposed to improve object detection and tracking by fusing information on the scene to catch tracking faults, such as occlusion, ID lost and motion blur. Other researchers [131, 122, 2] proposed new models to cope with UAV-based event detection both in inside and outside environments. They proposed ontology-based approaches to model knowledge on the scene and objects. Some approaches focus on a robust interpretation of events over time to abstract higher-level knowledge on a scene and provide refined descriptions of the whole scenario [140, 131, 103]. In [103], the authors propose a novel reasoning mechanism to deal with uncertainty in activity detection. In [140], the ontology-based model introduced in [123] is extended by considering a query-based temporal window to analyze spatio/temporal relations among tracked people and detect events over time. In [131] an ontology-based system, namely iKnow, detects activities of daily-living by merging dependencies among low-level and high-level concepts, such as locations and objects involved in activities. This model introduces the telicity criteria, which is applied to group already detected activities for situation interpretation. The approach, presented in Section 5.3, detects simple activities carried out by tracked scene objects, then, compositions of these activities over time enable the definition of higher-level complex activities. The knowledge modeling is achieved by ontology axioms and applying reasoning on them. The knowledge-based system proposed in [2] introduces a context layer over tracking, that employs an ontology composed of several sub-ontologies, each one devoted to a specific aspect/layer of the scene, from the lowest to the highest level (i.e., tracking data, scene objects, situations).

The approaches [123, 122, 144, 131, 2] employ knowledge-based methods to detect activities and situations, but they do not provide a methodological approach to achieve a scene description. This section introduces an ontology design pattern that provides the incremental steps (in form of ontological models) to describe a scene, at different levels of detail. Coding design patterns into ontologies has been proven to be useful for supporting and improving Semantic Web ontology engineering [146]. In [146], content-oriented patterns are shown to be useful to abstract knowledge and support composition. The next sections introduce a multi-ontology process design pattern to support knowledge acquisition and reuse about a UAV-taken scenario. The employment of a knowledge-based approach does not prevent the use of a statistical-based or probabilistic approach. In fact, in [147], ontologies and Markov Logic Networks are used synergistically to accomplish activity recognition.

Recent studies evidence the role of ontology for modeling the features arisen from the UAV-observed scene [123, 148, 149]. In [123], the ontology namely TrackPOI represents scene mobile objects (i.e., people, vehicles, etc.) and environments (roads, buildings, etc.) by starting from tracked scene data. Activity Ontology Design Pattern (ODP) [148] introduces a core ontology for activity modelling that can be used in different contexts. The activity is modelled along with its features (time duration, people involved, etc.). This ontology also allows the modeling of an activity as composed by simpler activities. An ontology similar to ODP is proposed in [150], the authors present a core ontology to model the activity and its features. Then, the model is extended with a specialization pattern and a composition pattern to, respectively, specialize the core ontology to model a specific domain and build complex activities from simpler ones. Situation Theory Ontology (STO) [149] concerns the modelling of concepts in Situation Theory (additional details will be provided in the next sections).

In the Situation Awareness domain, ontologies often combine classes modeling sensor-related information with classes modeling high-level features, such as relations among scene objects, events, and situations. The ontologies proposed in the literature are upper ontologies, representing general relations among the data, that can be specialized to accomplish a specific application. In [151], a novel method to knowledge representation for Situation Awareness is discussed. It uses RuleML-based domain theories and proposes the Situation Awareness (SAW) ontology. The ontology models a situation as a collection of goals, entities or objects and relations among these objects. The ontology also models events as acquired by sensors and allows the definition of dynamic representation over time by updating specific properties. The ontology is a core ontology, but its classes can be extended to represent situations occurring in specific domains. In [2], several connected upper ontologies are proposed to describe different aspects of the scene, such as tracked entities, scene objects, activities, etc.

Recalling the knowledge schema in Figure 5.2, that shows a methodological infrastructure to incrementally recognize objects, their activities and, systematically, describe a video frame scenario. The logic behind this schema needs solid formal modeling that finds its answer in the use of a thorough ontological design. Ontologies provide indeed formal models to describe axiom-based knowledge and infer new knowledge through semantic reasoning.

In this section a layered ontological model is discussed to achieve a synthetic scenario interpretation by starting from the video tracking data. The layered knowledge schema provides a methodological approach to yield a scenario description exploiting the ontology language. Bearing in mind one of the focal principles of the Semantic Web, viz., the data re-usability, the layered knowledge schema is achievable by integrating existing upper and domain ontologies, aligning similar concepts and extending them, in order to bridge different domain knowledge. Ontology integration is not an easy task to fulfil, due to the difficulties to relate distinct domains (ontology alignment). Poor ontology integration can result in excessive redundancy of information, with a consequent reduction in performance [152], that inevitably affect semantic reasoning and query processing [153].

To address this issue, this section introduces an ontology-based approach to incrementally model knowledge describing a real world scenario from a frame sequence of a video. The approach follows the schema in Figure 5.2 to achieve multi-granule knowledge generation: an incremental abstraction process of the video content, supported formally by ontology modeling. The knowledge is extracted for each layer is modeled as ontology concepts, corresponding to the main scene actors, and their relationships constituting movements, event, activities, and finally, situations on the scene. At each layer, data augments their expressive power (becoming higher level knowledge), thanks to the corresponding ontology model, that describes that level conceptualization. Applying the ontology-based reasoning on generated knowledge, the following upper layer of knowledge is inferred. The rest of this section details the ontological modeling at each layer, including Raw sensor data, Object, Activity and Situation layers. Then, the integration of the employed ontologies and the generation of knowledge. Finally, an experimental case study shows the functioning of the model.

5.4.1 Raw sensor data layer

This layer represents the basic level, namely, *0-layer*, to highlight the fact that it is an initial processing step, on which the ontological model is based. It indeed collects the input data from the UAV-recorded video, sensing the main actors of the scene and the environmental context. Video Analysis techniques are widely employed to accomplish this task: video tracking is performed to track the movements of the mobile scene objects, such as people, vehicles, etc.; also target classification information are returned about scene object identified.

The output is an XML-based file, described in Chapter 4, Section 4.3.1, including the information on the scene objects detected frame by frame. To detect the environment type, area classification is also provided for the types of ground areas present in the video. The classification results annotate each tracked object along with the area where they appear and the areas in its surroundings. In general, the XML file collects information types such as bounding boxes dimensions and positions, speed, direction as well as object identity and area classification, etc. The ontology modeling approach supposes that the generated XML file is the result of accurate video tracking as well as object recognition and classification activities, to guarantee an effective nested knowledge generation layering. Deep learning, as reinforcement learning are established techniques used in Video Analysis and represent a solid basis on which to build our ontological modeling.

The output results of *Raw sensor data Layer* are roughly the main mobile and fixed objects present in the scene, annotated with the class label. These data are the raw knowledge on the scene, on which our approach incrementally builds higher-level knowledge on the UAV-monitored scene.

5.4.2 Object layer: the thing object

As it has been demonstrated, the TrackPOI ontology provides the formal model to describe what appears in each frame, frame-by-frame. If a series of track instances, identified by a certain ID, appears in a frame sequence, it represents the same physical object. Moreover, in terms of ontology coding, the axioms related to the object presence in a time interval are replicated as many times as the number of frames is. To this purpose, the initial TrackPOI ontology has been extended with the $ThingObject$ class, that supports the conceptual abstraction of the object presence over time, by a digest, time-based axiom. Therefore, the $ThingOject$ class represents the scene object, moving in the scene over time, as a collection of all its bounding boxes ($Track$ instances) per frame. Figure 5.15 shows the class *ThingObject* that is related to the class *Track* by the relation *hasTrack*, or conversely, each *Track* is part of (*trackOf*) a *ThingObject*.

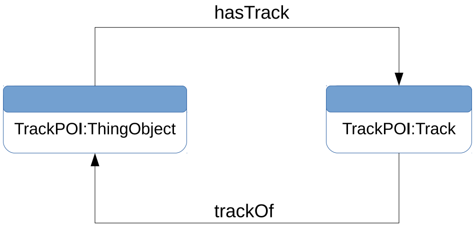


Figure 5.15: TrackPOI ThingObject class: the high-level dynamic object model

In other words, an instance of *ThingObject* is the actual object appearing in the scene, described by a sequence of *Track* instances (identified by the same ID) over time.

5.4.3 Activity layer: Ontology Design Pattern (ODP)

The activities carried out by the main actors in the scenes are modeled by using an ontology design pattern [148] (briefly, ODP) to model the common core of activities in different domains.

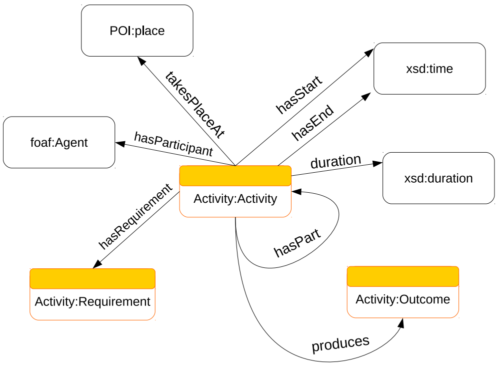


Figure 5.16: Activity OPD ontology for high-level activities modeling

Figure 5.16 shows the Activity ODP schema with classes and properties. According the schema in the figure, a generic activity has a starting and finishing time (respectively, described by the properties *hasStart* and *hasEnd*), represented by *xsd:time*; it lasts over time, the range of property *hasDuration* is *xsd:duration* which represents the activity time duration. Moreover, a generic activity can be composed of other activities. In fact, an activity individual, represented as an instance of the *Activity:Activity* class, can be related to its component activities through the *hasPart* property. The *Activity:Activity* class is connected by relations *Activity:hasRequirement* and *Activity:produces* to the two main classes that characterize the activity, the *Activity:Requirement* and *Activity:Outcome* classes, that represent the input and the output of the activity, respectively. These classes enable modeling logical order among the activities.

Classes from external ontologies are also used to contextualize the activity. Accordingly, in the figure, the *POI:place* class models the place where the activity occurred. The *foaf:Agent* class represents the participants in the activity.

The ODP ontology has been employed to model knowledge on detected activities (that specialize this generic class) and support the definition of higher-level complex activities.

5.4.4 Situation layer: Situation Theory Ontology (STO)

In common sense, a situation is often represented by a combination of circumstances in which someone or something finds itself or a specific status with regard to conditions and circumstances. A situation can be a simple people's activity, or the effect caused by some complex events. In Situation Awareness [7], situation is defined as the perception of some situational elements, the comprehension of their meaning and the projection of their state in the future.

The STO ontology models the fundamental concepts involved in the situation theory [149]. Situation theory concerns the situation semantics developed by Barwise and Perry [154, 155, 156] to reason over common-sense and real world situations. In this theory, a situation is composed of infons, elementary units of information that characterize a situation. More formally, it is defined on an n-ary relation R among n objects or individuals $a_1, \ldots, a_n$, therefore, it is written as follows: $\langle\langle R, a_1, \ldots, a_n, 0/1\rangle\rangle$. The infon represents a fact that can be true or false and it is represented by the last argument in the infon definition $(0/1)$ that expresses its own polarity. The relation (R) in the infon represents the type of event or action involving one or more individuals. The individuals $(a_1, \ldots, a_n)$ are entities (i.e., people, animals, etc.) that participate in the situation.

Figure 5.17 shows the core ontology schema of STO. The root class is *STO:Situation* which represents the situation. The classes *STO:ElementaryInfon*, *STO:Relation* and *STO:Individual* are involved in the situation definition. More specifically, the *STO:Situation* class is related to the *STO:ElementaryInfon* class

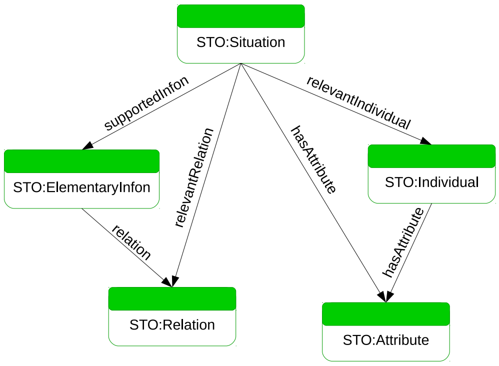


Figure 5.17: Situation Theory Ontology (STO): it models Situation Theory

by the *supportedInfon* relation. An *STO:ElementaryInfon* is an *STO:Relation* existing among one or more *STO:Individual* instances. The *STO:Attribute* class describes attributes that can be associated with both individuals and situations. The class is devoted to represent locations and time instants related to the situation or individuals.

5.4.5 Ontology Integration and knowledge generation

The ontologies are the building blocks of our layered knowledge scheme of Figure 5.2. They contribute to provide a high-level abstraction of the scene in a dynamic environment. Conceptual alignments or, more in general, portions of ontology merging and integration need to be harmonized in a comprehensive ontology model that reflects our schema.

Figure 5.18 shows the final ontology schema, with the integration model design (additional relations connecting the individual ontologies) in evidence. The figure strictly reflects the layered knowledge schema, namely from the bottom layer *Raw sensor data (layer 0), Scene object (layer 1), activity/event (layer 2), Situation (layer 3).*

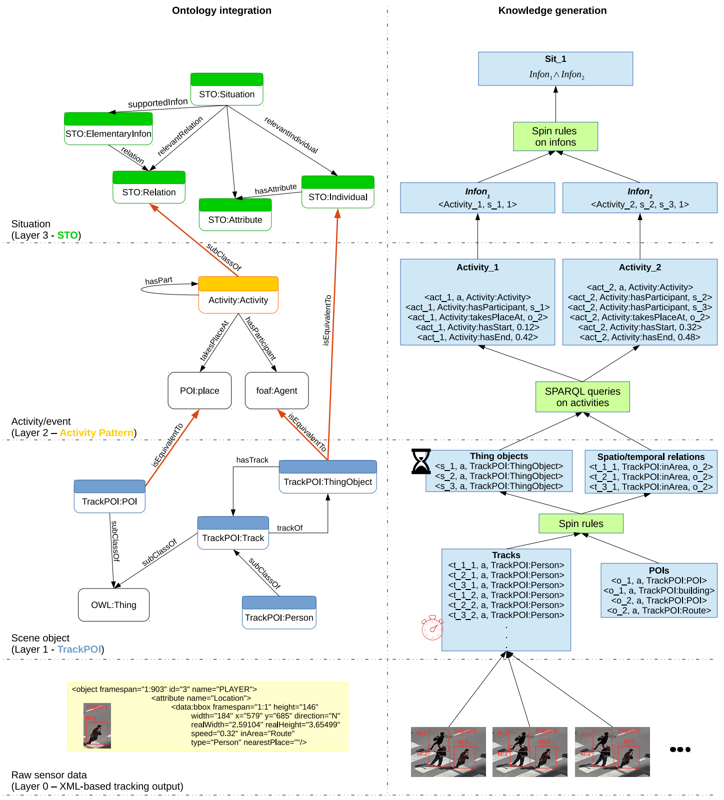
The layer 0 provides the xml-based data describing bounding boxes and their positions, as well as their membership class (for example, if the bounding box represents a person, a car, etc.), as described in Chapter 4, Section 4.3.

At the layer 1, the data, generated at the previous layer, are translated into semantic assertions that describe the recognized mobile and fixed objects as instances of the *TrackPOI:Track* and *TrackPOI:POI*, respectively, from the ontology TrackPOI. The track identifiers and class names are coded into semantic assertions: for example, the triple <*t*_1_2 *a* *TrackPOI:Person*> states that the track with *ID*:1 in the second frame (numbered as 2) represents a Person (in other words, *t*_1_2 is an individual of the class Person). POIs collected by Google Maps service, or detected by area classification at layer 0, are described by ontology assertions in a similar way.

Interactions between fixed (e.g., POIs) and moving objects are also identified in this layer. To this purpose, object positions, with respect to a specific area or just generic spatio-temporal relations occurring in the scene are detected. Therefore, triples representing spatio-temporal relations among tracks are generated. Furthermore, in this layer, the identification of the scene object, as composed of tracks appearing in a frame sequence, is accomplished as individuals of the *TrackPOI:ThingObject* class. Spin rules help the consolidation of the object movements and interactions, as well as the merging of the tracks associated to the same object (see Section 5.4.2 for details). For instance, the generated triple <*s*_1 *a* *TrackPOI:ThingObject*> represents the mobile scene object *s*_1 composed of tracks with ID equals to 1 from the video frame sequence, such as <*t*_1_1 *a* *TrackPOI:Person*>, <*t*_1_2 *a* *TrackPOI:Person*>, <*t*_1_3 *a* *TrackPOI:Person*>, etc.

In the layer 2, SPARQL queries are designed to elicit activities, that are based on the generated *TrackPOI:ThingObject* instances and spatio-temporal relations among tracks. In further details, according to the composition model presented in Section 5.3, queries support the detection of high-level activities over time [122]. The detected activities are represented as instances of the *Activity:Activity* class, then, new triples are generated. These triples relate the activity with the thing objects who carried out or participate to the activity and the place where it happened. In the figure, for instance, a generic activity *act_ 1* is characterized by the participant (the thing object named *s_ 1*) in that activity, the place where it occurs (the POI *o_ 2*) and the starting and ending times (at the second 0.12 and 0.42, respectively).

Let us notice that the layer 1 and layer 2 are joined by new additional relations ($isEquivantTo$), that connect similar concepts from the ontologies TrackPOI and ODP, respectively.


Ontology integration
Knowledge generation
STO:Situation
supportedInfon
STO:ElementaryInfon
relevantRelation
relation
relevantIndividual
STO:Relation
STO:Attribute
hasAttribute
STO:Individual
Situation
(Layer 3 - STO)
subClassOf
hasPart
Activity:Activity
takesPlaceAt
hasParticipant
isEquivalentTo
POI:place
foaf:Agent
Activity/event
(Layer 2 – Activity Pattern)
isEquivalentTo
isEquivalentTo
hasTrack
TrackPOI:ThingObject
TrackPOI:POI
subClassOf
TrackPOI:Track
trackOf
subClassOf
subClassOf
OWL:Thing
TrackPOI:Person
Scene object
(Layer 1 - TrackPOI)
Raw sensor data
(Layer 0 – XML-based tracking output)
Frame 1
Id:3
Sit_1
$Infon_1 \wedge Infon_2$
Spin rules
on infons
$Infon_1$
<Activity_1, s_1, 1>
$Infon_2$
<Activity_2, s_2, s_3, 1>
Activity_1
<act_1, a, Activity:Activity>
<act_1, Activity:hasParticipant, s_1>
<act_1, Activity:takesPlaceAt, o_2>
<act_1, Activity:hasStart, 0.12>
<act_1, Activity:hasEnd, 0.42>
Activity_2
<act_2, a, Activity:Activity>
<act_2, Activity:hasParticipant, s_2>
<act_2, Activity:hasParticipant, s_3>
<act_2, Activity:takesPlaceAt, o_2>
<act_2, Activity:hasStart, 0.32>
<act_2, Activity:hasEnd, 0.48>
SPARQL queries
on activities
Thing objects
<s_1, a, TrackPOI:ThingObject>
<s_2, a, TrackPOI:ThingObject>
<s_3, a, TrackPOI:ThingObject>
Spatio/temporal relations
<t_1_1, TrackPOI:inArea, o_2>
<t_2_1, TrackPOI:inArea, o_2>
<t_3_1, TrackPOI:inArea, o_2>
Spin rules
Tracks
<t_1_1, a, TrackPOI:Person>
<t_2_1, a, TrackPOI:Person>
<t_3_1, a, TrackPOI:Person>
<t_1_2, a, TrackPOI:Person>
<t_2_2, a, TrackPOI:Person>
<t_3_2, a, TrackPOI:Person>
.
.
.
POIs
<o_1, a, TrackPOI:POI>
<o_1, a, TrackPOI:building>
<o_2, a, TrackPOI:POI>
<o_2, a, TrackPOI:Route>
Id:1
Frame 1
Id:3
Id:2
Id:1
Frame 2
Id:3
Id:2
Id:1
Frame 3
Id:3
Id:2
•••

```
<object framespan="1:903" id="3" name="PLAYER">
        <attribute name="Location">
                <data:bbox framespan="1:1" height="146"
                        width="184" x="579" y="685" direction="N"
                        realWidth="2.59104" realHeight="3.65499"
                        speed="0.32" inArea="Route"
                        type="Person" nearestPlace=""/>
```

Figure 5.18: The integrated ontology model and an example of knowledge generation at each layer. The thick coloured lines represent the properties that integrate the three ontologies.

More specifically, the *TrackPOI:ThingObject* instance is the high-level object that carries out the activity; since it represents the main participant of the activity, it is equivalent to the *foaf:Agent* class. In this way, through the *Activity:hasParticipant* property (that connects the *Activity:Activity* class to the *foaf:Agent* class), the activity (i.e., *Activity:Activity* instance) is related to the object doing it (i.e., *TrackPOI:ThingObject* instance). Similarly, the *TrackPOI:POI* and *POI:place* classes are equivalent and related to the *Activity:Activity* class through the property *Activity:takesPlaceAt*.

At layer 3, the high level ontology STO is in charge of situation description. Figure 5.18 shows the connection between the STO and the ontologies in the two underlying layers. As stated in Section 5.4.4, the *STO:Individual* class, in the STO ontology, models entities (i.e., people, animals, etc.) that carry out activities or are involved in events and situations. The *TrackPOI:ThingObject* class represents the same concept (i.e., it is assumed to be equivalent) to *STO:Individual.* The *Activity:Activity* class exclusively represents activities carried out by one or more scene objects. Activities are also modelled in the STO ontology by the *STO:Relation* class. The *Activity:Activity* class is designed as a subclass of the *STO:Relation* class, that connects directly the ODP ontology to the STO ontology.

When new *Activity:Activity* instances have been generated at layer 2, the same instances are also of type *STO:Relation.* At layer 3, infons on each generated *STO:Relation* instance are produced. Precisely, an instance of *STO:ElementaryInfon* is yield, for each detected activity type in *Activity:Activity*, equivalent to *STO:Relation.* These instances represent the detected activities along with time, location and the participants to the activity. Concatenations of infons defined by Spin rules allow defining high-level situations. For instance, given the infons $Infon_1$, $Infon_2$ and the situation Sit_1 defined by the rule $R : Infon_1 \wedge Infon_2 \implies Sit\_1$; if the two infons $Infon_1$ and $Infon_2$ are generated, the rule R allows the detection of the situation Sit_1.

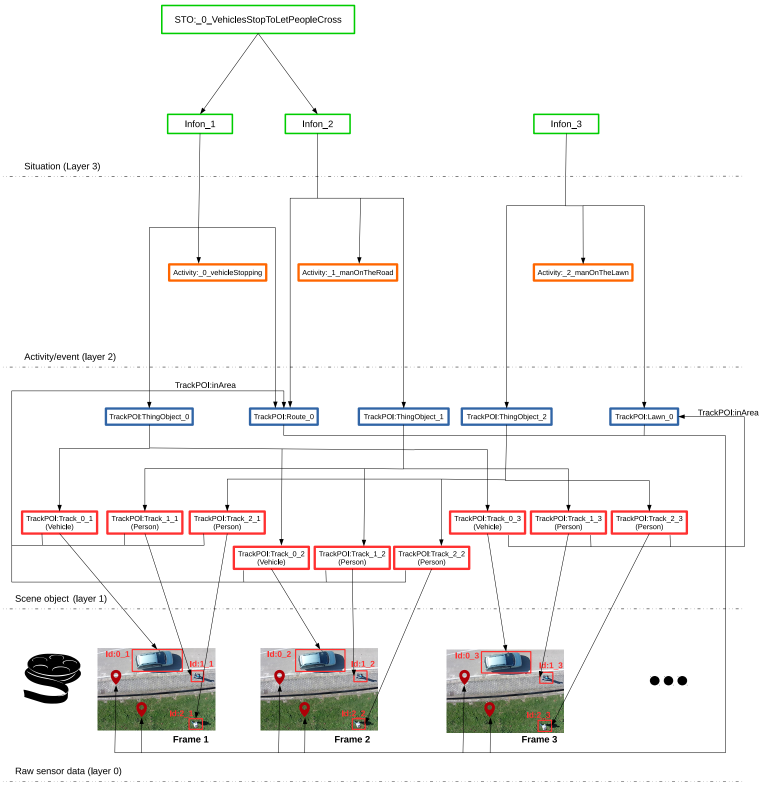


Figure 5.19: Knowledge augmentation through the ontology-driven layered schema: an illustrative example

5.4.6 Representation of a case scenario

This section presents a case study showing the applicability of the proposed ontology modeling and effectiveness in the scene description, on a real-world video. Figure 5.19 shows the generation of the ontology population, through the layers of the knowledge schema, starting from the initial raw data to yield a high level description of the scenario. The video frames, at the layer 0, show a typical outside scenario recorded by a camera-equipped UAV. A vehicle is running while a person is crossing and another person is walking on the lawn beside the road. As stated, data retrieved by sensors and tracking algorithms allows us to recognize targets in the scene. The tracking algorithm used in this case study estimates camera movements for background scene extraction and identifies object position. Moreover, feedforward control [123] has been used to improve trajectory tracking of objects through frames. In the example, the tracking algorithm returns the objects identified by *id:0*, *id:1* and *id:2*. Then classification algorithms have been employed to object and background area annotations. The used object classifier considers three object categories: people, vehicles and unknown objects. The object classification is performed frame-by-frame and, then, the object label is got through a majority voting approach [123]. The classification results are used to annotate each detected scene object, adding a class-type field, expressing its identity. Identity and area annotations on scene objects are added as attributes to tags, expressing the tracked objects, in the original tracking output file. The area classifier detects the main background environments (e.g., lawn or road) where the objects stay or places they get close to [122].

Identity and area annotations on scene objects are added as attributes to tags, expressing the tracked objects, in the original tracking output file. Tracking and classification data are then encoded into ontology assertions [123], generating actual instances of TrackPOI ontology. At layer 1, for each frame, the instances of Vehicle and Person are created. In the frame numbered 1, the generated instances *TrackPOI*:*Track*_0_1, *TrackPOI*:*Track*_1_1 and *TrackPOI*:*Track*_2_1, represent the tracks produced at the layer 0 and are individuals of TrackPOI ontology *TrackPOI*:*Vehicle*, *TrackPOI*:*Person*. Considering video frames, it is possible to seek the same track through frames.

Tracks with the same ID are grouped in a unique dynamic entity (i.e., thing object) representing the mobile object in the scene. For instance, the instances *TrackPOI*:*Track*_1_1, *TrackPOI*:*Track*_1_2 and *TrackPOI*:*Track*_1_3 represent the tracks with the ID equals to 1 in frames 1, 2 and 3, respectively. These tracks, representing the same instance of the *TrackPOI*:*Person* class through the frames, are grouped to build the *TrackPOI*:*ThingObject*_1 instance of the class *TrackPOI:ThingObject*. At the same time, the generated *TrackPOI*:*Track* instances are related to *TrackPOI*:*POI* instances, representing the environments where they move, through the *TrackPOI:inArea* property. Through this property, tracks of the vehicle and the person with *ID*:1 are found in the area of the route, while the other person with *ID*:2 is found on the lawn besides the route. These spatial relations are also timed because related to a specific frame. Therefore, the generated spatio/temporal relations support the contextualization of the object movements and interactions with other objects. The outcome of layer 1 is the identification of three objects (belonging to the class *TrackPOI: ThingObject*), and their relation with the places where they appear (i.e., the route and the lawn).

At the layer 2, some rules are designed on the *TrackPOI*:*ThingObject* instances and the spatio/temporal relations. Collecting data on objects and their spatio-temporal relation, by SPARQL reasoning, activities are detected. In the figure, some specialized activities are shown: they are carried out by the two people and the vehicle arisen at layer 2 of Figure 5.19. More precisely, the following activities are elicited: *Activity:_0_vehicleStopping*, *Activity:_1_ManOnTheRoad*, *Activity:_2_ManOnTheLawn*. At high level of description, the observed scenario shows a vehicle which is stopping (*Activity:_0_vehicleStopping*) when the person crosses the route (*Activity:_1_ManOnTheRoad*). Then, the other person is simply walking in the lawn area (*Activity:_2_manOnTheLawn*).

```
SELECT ?ob ?track ?time ?poi
WHERE {
?track a trackpoi:Person .
?track trackpoi:inArea ?poi .
?poi a trackpoi:Route .
?track trackpoi:hasTime ?time .
?track trackpoi:track_ID ?id .
?track trackpoi:trackOf ?ob .
} ORDER BY ?id ?time
```

Listing 5.4: *manOnTheRoad* activity: SPARQL query for detecting people on the road

As a SPARQL query example for activity definition, let us consider the query to detect the activity instance *Activity:_1_ManOnTheRoad* shown in Listing 5.4. The SPARQL query detects people walking on the road over video time. This query makes possible to create an instance of a specific class *Activity:ManOnTheRoad*, subclass of *Activity:Activity*, for each track who carried out this activity. The query returns a list of tracks ordered by their ID and time when they appear in the video. The *TrackPOI:trackOf* property supports the identification of the person (*TrackPOI:ThingObject* instance) walking on the road, while its track time serves the detection of the times of entrance and exit on the road.

At the layer 3, the scene description becomes concise, and reaches a very high level of abstraction. Situation Theory is applied to the detected activities and scene objects to abstract knowledge from them and provide high-level situations describing the whole scene. Infons are generated on the detected activities and scene objects to relate all the information and build situations. The situations are Spin rule-defined as concatenation of infons. The outcome of the layer 3 are the infons *Infon_1* and *Infon_2* in correspondence with activities *Activity:_0_vehicleStopping* and *Activity:_1_ManOnTheRoad*, respectively. The Spin rules define a situation, namely *STO:_0_vehicleStopToLetPeopleCross*, that comes from the concatenation of these infons, in the road context. This situation exactly captures the main action happening in the road scenario, and provide a human-oriented, high-level view of the scene.

The proposed ontology modeling provides a systematic way to feed a knowledge base describing a video, ranging from the identification of the individual objects to the occurring activities, till to incrementally achieve a general, high-level scenario description.

In order to assess the applicability of this approach and its effectiveness in term of scenario description, some videos have been processed, as described in the case study. Three videos[9] recorded in our campus have been processed: they show people and vehicles carrying out some activities in different environments, such as roads, lawns and heliports. Table 5.5 shows the results of the application of the proposed ontology model, according to the multi-layer knowledge schema. The table provides the video

[9] `https://tinyurl.com/yygg282c`

Table 5.5: Situations and activities recognized in the videos.

Video	Situations	Activity	TO	Type	POI	Start	End
Video #1	Sit_0_ObjectNearer	0_ObjectNearer	TO_0	Person		00:00:00	00:01:15
	Sit_1_Grouping	5_Grouping	TO_3	Person	Lawn1	00:00:40	00:00:45
		6_Grouping	TO_1	Person	Lawn1	00:00:40	00:00:45
	Sit_2_ObjectNearer	4_ObjectNearer	TO_3	Person	Lawn1	00:00:12	00:01:11
	Sit_3_ObjectNearer	3_ObjectNearer	TO_2	Vehicle	Route1	00:01:09	00:01:12
		3_ObjectNearer	TO_1	Person	Route1	00:01:09	00:01:12
	Sit_4_Grouping	0_Grouping	TO_1	Person	Lawn1	00:00:15	00:00:24
	Sit_5_Grouping	4_Grouping	TO_3	Person	Lawn1	00:00:23	00:00:25
Video #2	Sit_3_ManCrossing	0_ManCrossing	TO_1	Person	Route1	00:00:50	00:00:55
	Sit_1_Grouping	0_Grouping	TO_3	Person	Lawn1	00:00:58	00:01:00
		1_Grouping	TO_1	Person	Lawn1	00:00:58	00:01:00
	Sit_2_Grouping	3_Grouping	TO_3	Person	Lawn2	00:00:29	00:00:31
		2_Grouping	TO_1	Person	Lawn2	00:00:29	00:00:31
	Sit_0_VehicleStops-toLetPeopleCross	4_Stopping	TO_2	Vehicle	Route1	00:00:50	00:00:55
		1_ManOnTheRoad	TO_1	Person	Route1	00:00:50	00:00:55
Video #3	Sit_2_ManMoving InTheHeliport	0_ManMoving InTheHeliport	TO_0	Person	Heliport	00:00:09	00:01:00
	Sit_3_ManMoving InTheHeliport	1_ManMoving InTheHeliport	TO_3	Person	Heliport	00:00:18	00:00:46
	Sit_4_Grouping	1_Grouping	TO_2	Person	Heliport	00:00:05	00:00:16
	Sit_5_Grouping	0_Grouping	TO_1	Person	Heliport	00:00:05	00:00:16
	Sit_0_ObjectNearer	4_ObjectNearer	TO_2	Person	Heliport	00:00:00	00:00:16
		5_ObjectNearer	TO_3	Person	Heliport	00:00:00	00:00:16

Detected situations (Situations)
Activities leading to situations (Activity)
Thing Object (TO)
Object identity (Type)

POI where activities happened (POI)
activity starting time (Start)
activity starting time (End)

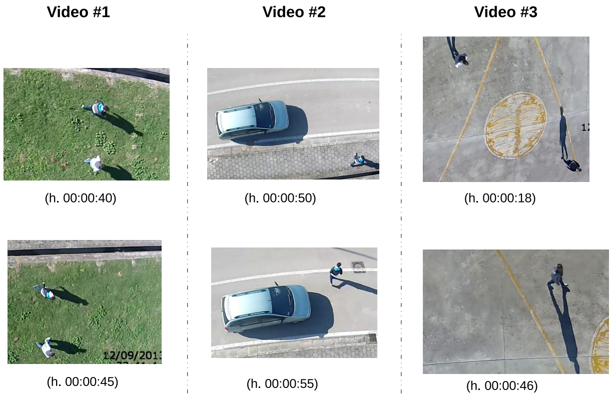


Figure 5.20: Situation detection: frames from Video #1: people grouping; Video #2: people crossing; Video #3: people moving on a heliport

content description: specifically, for each video, the situations and the activities that compound these situations are shown, in the time interval they occur. Then, each activity includes the thing object who carried out the activity, the thing object type, the POI where the activity happened and the activity beginning and ending times. Figure 5.20 shows one of the situations recognized in each of the three videos (i.e. people grouping from *Video #1*, people crossing from *Video #2*, people moving on a heliport from *Video #3*). Situations are described exactly by the time interval they occur, expressed by the starting and ending frames. Let us notice that by comparing situations, objects and times in the figure with the table results, the detected situations correspond to those found in the videos.

For instance, looking at *Video #2* in Table 5.5 the recognized situations are *Sit_ 3_ ManCrossing*, *Sit_ 1_ Grouping*, *Sit_ 2_ Grouping*, and *Sit_ 0_ VehicleStopstoLetPeopleCross*. The *Video #2* shows a road scene with people grouping, and a crossing happening in presence of an oncoming vehicle (see Figure 5.20). In Table 5.5, for *Video #2*, the situation *Sit_ 3_ ManCrossing* is produced by the individual activity *0_ ManCrossing* (in the *Activity* column); the situations *Sit_ 1_ Grouping*, *Sit_ 2_ Grouping* are described by the grouping activities (identified as *0_ Grouping*, *1_ Grouping*, *2_ Grouping*, *3_ Grouping*). Each grouping activity can be carried out by only one thing object, so there are as many grouping activities as there are thing objects involved in the grouping. The thing objects namely *TO_ 3* and *TO_ 1* are both recognized as persons (*Type* column) and participate to the situations *Sit_ 2_ Grouping* and *Sit_ 1_ Grouping.* More interesting is the situation *Sit_ 3_ VehicleStopstoLetPeopleCross* that represents a vehicle stopping to let people cross the road, described by the activities *1_ ManOnTheRoad* and *4_ Stopping.* The two activities involve two thing objects recognized as a person (*TO_ 1*) and a vehicle (*TO_ 2*). Situations and activities last a certain amount of time, from a starting to ending time (*Start* and *End* columns, in the table).The starting and ending times allow us to describe the temporal succession of the situations detected in the video. The *Video #2* indeed shows initially two people grouping (*Sit_ 2_ Grouping*), then moving away from each other, and one of them crosses the street (*Sit_ 3_ ManCrossing*) while an oncoming vehicle stops to let the person cross (*Sit_ 0_ VehicleStopstoLetPeopleCross*); in the end, the people meet again (*Sit_ 1_ Grouping*) (see Figure 5.20).

5.5 Discussion

The activity composition model, presented in Chapter 5.3, bridges Computer Vision and Semantic Technologies, to UAVs to achieve a high-level video comprehension. 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 novelty of this model is the object activity modeling 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. A UAV providing descriptions of high-level articulated activities over time can support human operators, employed in surveillance and monitoring of various environments, with human-like detailed video content analysis.

The approach, discussed in Chapter 5.4, presents a systematic ontology-based design process based on the introduced multi-layer knowledge schema, that composes the scene increasingly at a high level of abstraction. The layered knowledge model indeed allows feeding knowledge on the scene incrementally, from tracked data to the situations describing the scene. The integrated ontology model exploits the features of several well-known ontologies to thoroughly model different aspects of the scene and achieve complete scene comprehension. Data tracking along with activity and situation (theory) modeling support the three levels underpinning the Situation Awareness: *Perception* (collecting row sensing data), *Comprehension* (seeking main actors in the scene: e.g., objects and carried activities), *Projection* (assessing possible critical issues on the detected situations).

The proposed ontology design is a kind of guideline that, reflecting the multi-layer knowledge schema, produces a formal knowledge modeling as well as arise the semantic description on an observed scene. In the light of the recent literature on situation comprehension, the main benefits of the proposed approach are briefly listed below.

- **An ontology design pattern for scenario understanding.** The whole ontology can be considered as a sort of ontology design pattern, coming from the modeling and integration of ontologies intended to portray the layering of our proposed knowledge schema described in Figure 5.2. In particular, the ontologies ODP and STO are indeed ontology design patterns, in charge of covering the *Activity* and *Situation* layers, respectively. The *Object* layer is the only one achieved with a domain ontology, and, for this reason, it can be easily replaced with another ontology, if a different video context (for example, the video scenes take place in a environment other than a road scenario) appears.

- **A modular design process for easy methodological integration.** The ontology design not only offers seamless extensibility at the ontology design level, but the modular layering also guarantees high flexibility and interchangeability of the methodological approaches for target tracking and classification in the *Raw sensor data* layer. The employment of high-performance Machine and Deep Learning methods for target tracking and classification tasks, for example, can enhance the effectiveness of the global system. Depending on the computer vision methods, used in the *Raw sensor data* layer, the ontology model can combine/compound more or less accurately detected scene objects, in order to produce higher-level scene descriptions.

- **A knowledge base to support video content analysis.** The ontology model allows populating a knowledge base describing the video content, collecting, depending on the layering of the knowledge schema, the information granule associated with the corresponding knowledge layer. The knowledge base is accessible by SPARQL queries: objects, activities, and situations appearing in a video (or in a portion of it) can be recovered by a query easily. The collected knowledge becomes a flexible repository to facilitate video content analysis targeted, for instance, at surveillance and monitoring applications.

- **A human-oriented scenario description.** The role of semantics is crucial in the scenario description: modeling a situation as a composition of activities and, in turn, an activity as spatio-temporal relations among objects and between the object and the environment, enables the logical "thinking" process, for understanding what really is happening in a scene and explaining why particular conclusion is achieved. The logics behind a situation can yield human-like video content description along with the reasoning steps that build a situation.

The proposed approach provides a semantic support for object detection and scenario description, if used in combination with Machine and Deep Learning methods, whose synergy provides solid performances.

Chapter 6

Multi-UV systems for scenario interpretation

6.1 Introduction

The goal of surveillance systems is to perform the real-time monitoring of persistent and transient objects within a specific environment: from the sensors to the final output, they support data gathering to processing, transform the information into knowledge through inference capability and, then, enhance situation awareness for decision-making tasks. The first objective of these systems is to assess the situation automatically: they offer a comprehensive understanding of scenes and their evolutions, especially to interpret the actions and interactions of the observed objects.
The situation understanding is a complicated activity which includes the acquisition of the initial raw data collected by different environmental sources (sensors, video, etc.) toward an incremental enrichment and aggregation (data fusion) to generate final information [157, 158, 83].

Multiple and heterogeneous sensing sources provide a different field of view of the same scene, not just for improving robustness and monitoring performance of the whole system, but also for a reliable and feature-rich perception of the current evolving situation, and the consequent possible decision to take (Figure 6.1). Each source individually represents a subsystem targeted at performing specific tasks, providing accurate and precise details about the tracked target. Then, data from the different sources are collected and merged together by a collector (often a ground control station), to provide a comprehensive perception of the acquired scenario. Contextual data (relation between moving objects, stationary-moving objects, stationary objects) are often taken into account, in order to elaborate also a wider awareness of the comprehensive scenario.

In order to fully monitor the environment and acquire meaningful data, the surveillance of outside scenarios requires mobile devices, capable of monitoring targets and better depict the whole environment.

6.1.1 From Unmanned Ground Vehicles (UGVs)...

Remote reconnaissance and environment monitoring is historically in charge of Unmanned Ground Vehicles (UGVs) that often participate in collaborative tasks for detection and tracking [159]. The UGVs deployed in a given environment can perform repetitive tasks with precision, efficiency and reliability; they are often targeted at a specific task, to reduce the design complexity. At the same time, the performance of an unmanned ground robot is crucial to assess its capability in obstacle detection. Collision avoidance is a challenging problem in UGV navigation since path planning and navigation algorithms rely on the obstacle's profile and their spatial distribution [160]. UGV navigation includes the surrounding environment perception, to identify existing obstacles and paths, by a path planning: an ordered sequence of intermediate points that the UGV must visit and reach them to generate a collision-free path from origin to destination, and a control of UGV actions and movements, to guarantee that UGV follows the right path [161]. A UGV collects information about the spatial distribution of obstacles from its surroundings by employing environment perception sensors, such as laser scanner, infrared, sound navigation and ranging (SONAR) and cameras etc.

One of the main problems encountered during the deploy of a UGV in rough environments is the limited field of view obtained by the on-board cameras and sensors. In very hazardous environments, such as those that concern with demining operations, it can be really hard to be aware of the situation and make a decision on the best path planning. Moreover, the sensing feature has a non-derisory cost, for instance, the laser scanner for distance measurement can be quite expensive in terms of power consumption and cost [162].
Unmanned systems have to offer a well-balanced relationship between the quality of support, reliability and additional workload; thus the synergistic use of UGVs and UAVs is often the best solution for improving navigation capabilities of multi-sensor situation-aware systems [159] whose purposes and applications range from real-time surveillance, entertainment, defense, military, and delivery.

6.1.2 ... to Unmanned Aerial Vehicles (UAVs)

Unmanned Aerial Vehicles (UAVs) represent a clear, low cost reply to (ground-plane) surveillance systems: they support the object detection and tracking [70, 71, 83], providing a complete description of the scene. Surveillance and coverage of a dynamically changing environment is an important task for which a UAV can be deployed; enabling a UAV to an intelligent visual surveillance is a very useful and desirable capability, that basically involves some stages such as moving object definition, recognition, tracking [163], behavioural analysis [164], and retrieval. These stages are accomplished by formal and methodological approaches in the area of machine vision, clustering [165] and pattern analysis [166], artificial intelligence [4] and data management that contribute to define and model the UAV situation awareness (SA).

The increase of the UAV awareness level consequently raises up its autonomy level. Therefore, individual unmanned vehicles can communicate, coordinate and finally interact with each other, to yield collaborative teams of unmanned assets. Coordinate activities indeed improve the effectiveness of coalition unmanned systems through the acquisition of the raw data from different environmental sources (sensors, video, etc.) and the incremental enrichment and data aggregation (data fusion) towards "active" information, i.e., the knowledge that describes the observed area. The primary goal of the unmanned vehicle systems is to support the automated situation assessment: a comprehensive understanding of the scenes and their evolutions, especially the actions and interactions of the fixed and mobile objects appearing in the scenario [157, 158, 83]. Multiple and heterogeneous sensing sources provide a different field of view of the same scene, not just targeted at improving robustness and monitoring performance of the whole system, but at providing a reliable and feature-rich perception of the actual scenario, and then the consequent decision to take in view of its feasible evolution [167]. Figure 6.1 shows a cooperative system of unmanned vehicles: each one individually represents a subsystem targeted at performing specific tasks, providing precise details about the tracked target. Data from the different sources are collected and merged together by a collector (often a ground control station), which processes them to provide a comprehensive perception of the acquired scenario.

6.1.3 Multi-sources fusion

Intelligent monitoring, detection and control are becoming hot topics in many safety-critical application domains, such as fire detection, traffic congestion or accidents, etc. It would be highly desirable that unmanned vehicles exhibit human-like behaviours. For example, in aerial video surveillance, a UAV after having acquired data from sensors, video and context, should be able to merge the acquired data and thanks to some cognitive inference-based ability, elaborates them, in order to get its own situation assessment observing the evolving scene.

Generally, the data fusion focuses only on grouping detection and presentation of critical events, while the threat identification is just the outcome of a predictive analysis, leaving the final decision to human experts [168]. Automated situation assessment and scene understanding are the uncompromising desiderata of every situation awareness system. Many approaches in machine learning [169], probability theory [170], fuzzy inference [171], Markov Logic Network [172] try to trace patterns for isolating the threatening behaviour, intrusion detection, traffic analysis, event and state-based detection. One crucial requirement for an effective situation awareness is the acquisition and integration of information at multiple scales [173]. A significant challenge for such systems is indeed to collect data from heterogeneous distributed sources [157] and then combine them to compose a richer data level, also called *information level* that is the main avenue to enhance the situation understanding. Multi-sensor data fusion as well as extracted contextual information enhance the knowledge about entities and objects involved in a scene. Data fusion produces an intermediate stage that adds relevant details in the description of events, scenes, situations. This new knowledge on the data provides a better view on what is shown in the scene, which events happened and which situation is occurred: it is the background layer to deduct, by inference, new information that provides a higher abstraction level of the comprehensive scene. The acquired data, and then the processed knowledge become awareness: the system becomes "aware" about the evolving scene, can recognize the criticality level of the situation, and then knows how to act accordingly. An efficient situation awareness system must acquire data from many sources through the use of real time video analysis, multiple object models, and pattern analysis, and then process them, in order to provide comprehensive situation understanding. For example, a camera captures a video about a truck and some smoke comes out of the truck (event); in meanwhile a sensor in the neighbouring area detects smoke in the air; a such system should collect the data from the two sources and then deduct that the truck probably has a malfunction to its engine (alerting situation); at that point, the system may decide whether or not to call the roadside assistance.

6.2 UVs as intelligent agents

Multi-agent architectures well suit to model unmanned robot systems, generally employed in patrolling, surveillance, search and rescue and human-hazardous missions. The agent paradigm indeed encapsulates some specific features that are the basic requirements of unmanned vehicles. An agent is able of performing autonomous actions, in order to meet its design goals; it is proactive: it exhibits a goal-directed behavior and takes its own initiatives, in response to the environment where is placed. In team-work design, it achieves collaborative and cooperative activities to reach the collective goal. The agent-based paradigm acts as a glue among heterogeneous unmanned systems communicating through distributed asynchronous interactions. Thanks to their intrinsic features such as simplicity, flexibility, responsiveness, self-organization, cooperation/coordination, low-cost agent design, they can be applied to a wide range of applications, from tackling complex problems [174], the cooperation of agents-robots [175], till to data fusion

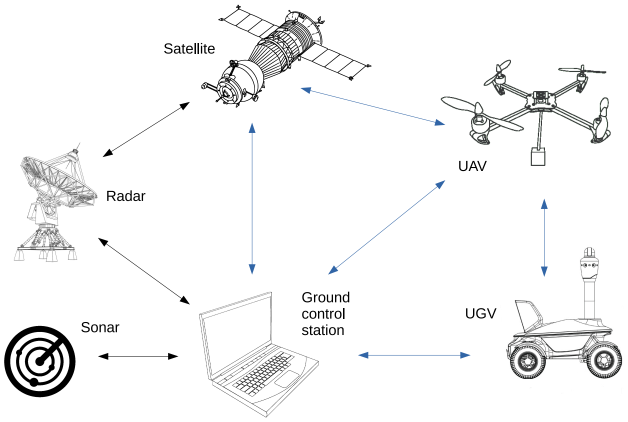


Figure 6.1: Multi-sensing unmanned vehicle systems for a coordinate data processing in dynamic environments [167]

and game-solving approaches [176]. Multi-agent systems provide a technology supporting the fusion of several traditional Unmanned Aerial Vehicle system areas: autonomy and navigation, attitude control, telemetry, etc. Moreover, they provide a distributed architecture, rejecting sequential top-down programs and preferring simple, distributed and decentralized processes with a direct access to sensors and actuators of the agent-robot. The effectiveness of the multi-agent systems is also due to the possibility to encapsulate a reasoning model in the agent-based paradigm: the agent exhibits some deductive capabilities that enable it to make decisions. The multi-agent modeling is a consolidated paradigm to support distributed systems that cooperate to reach a final objective. Thanks to these features, it is suited for designing Unmanned Vehicle systems, which are often mission-oriented, with requests for control, communication and coordination mechanisms. The next sections present an agent-based model to lead UVs to scene comprehension.

6.2.1 An agent-based multi-UV system

An agent-based UV system for scene comprehension has been designed. Figure 6.2 depicts its functioning through a logical schema, that represents the whole model as composed of three main functionalities: *Agent-based Knowledge Collection*, *Data Fusion-based Awareness* and *Intervention*. The main objective of this approach is the design of a team composed of multiple UVs, where each UV is modelled as an agent, namely Vehicle Agent (V. Agent, in figure). The agent-based UV team analyses scenes from an environment, where the individual vehicles stay, and each Vehicle Agent generates a comprehensive reading of a scenario portion, depending on the viewpoint, type and features of the UV. The design of a multi-agent system suits to model UVs of different type (i.e., UAV, UGV). Therefore, each agent controls a vehicle that collects data from the environment, hence the agent is instantiated to process the collected data. Then, depending on the vehicle feature, the agent processes the data and codes them into semantic statements, that become the initial concepts. The *Agent-based*

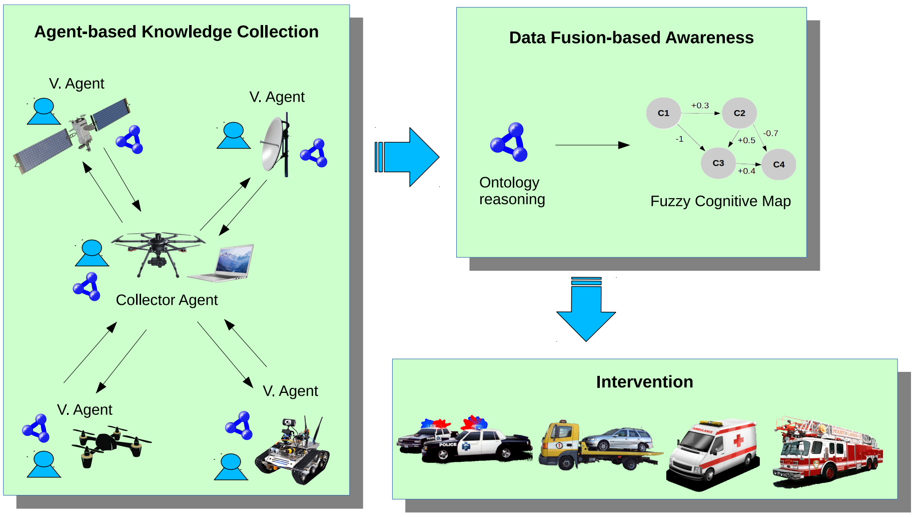


Figure 6.2: Model overview: Unmanned Vehicle Systems (UVs) are modeled as independent Vehicle Agents (V.Agent), the Collector Agent fuses other agent output to evaluate the most suited kind of intervention.

Knowledge Collection concerns the collection of data acquired from the environment by the vehicle agents through their sensors, cameras, etc. After the initial processing, thanks to the ontological coding, data becomes knowledge. One of the agents in the team is called the "collector", and it is assigned with the management of the *Agent-based Knowledge Collection* process. The collector agent could govern a UAV or a mobile UGV equipped with sensors, or even a simple laptop, enhanced with a cognitive model to gather data and conceptualizations provided by other agents and generate new inferred knowledge. The agents can be mobile or fixed devices, such as sensor-equipped UAVs, UGVs, or even satellites, infra-red cameras, radars or gas sensors. The agents are task-oriented: this means that, equipped with specific sensors, they can get data about the scene objects or features of the environment they are observing.

The collected raw video data are processed by each vehicle agent, and, then, transformed into semantic concepts according to an ontological model. For instance, data about a scene object, that has been detected and recognized as a car by a Computer Vision algorithm, is coded as the ontological concept *Car*. After that the agents have accomplished their own tasks, they provide the collector-agent with the conceptualization they produced.

The collector-agent employs a mental model, defined as a global Fuzzy Cognitive Map (FCM), which represents high-level knowledge on the main ground area where the scene, monitored by the agent vehicles, evolve. The area description is extended with the geo-referential data and the fixed objects. The mental model allows the collector to elicit events and state what is happening in the observed scene. If the collector evaluates the data reported by an agent partial or unsatisfying, it can query that agent or even other agents, in the team, asking for data about a precise geographical position. The results, provided by the agents, are taken as initial concept values to initialize the FCM, and, depending on the returned concept values (i.e. "people presence", "fire detection"), some portions of the FCM are activated and run. The FCM simulation generates new global, high-level knowledge about what happened in the observed ground area thanks to the individual vehicle agents, which generate semantic-coded data, describing portions of that area. This way, the collector agent builds a global view of events occurred in the observed area and it can make decisions according to the final concept values returned by the FCM simulation. Consequently, on the basis of the comprehensive situation assessment, the agent-based system can decide if and which rescue intervention is required, among firemen, police, ambulance, highway patrol, etc. Moreover, the FCM simulation process not only highlights the situations, but it also determines the main possible causes that lead to alerting events.

6.2.2 Vehicle agent knowledge collection

In the *Agent-based Knowledge Collection*, the designed vehicle agents can control vehicles of different type (i.e. UAVs, UGVs, satellites, etc.), and according to the vehicle features (environment-installed sensors, satellite-based services, servers, smart robots or simple laptops), they can be provided with different functionalities. Therefore, agent vehicles can have just basic functionalities, such as sensor data acquisition, or more advanced capabilities, such as target detection and identification, scene object activity recognition, environmental context definition, alerting event detection (i.e. conflagration, earthquake, seaquake detection).

Each agent is assigned with a task, hence its main goal is to solve the assigned task independently from the other agents. To this purpose, the vehicle agent is equipped with proper methodologies to accomplish its task, such as Video Analysis and Machine Learning (ML) methods to achieve target detection. For instance, an agent can control a UAV, equipped with gas sensors, to individuate city areas affected by pollution, or a UGV, provided with cameras and ML classification algorithms designed for people detection in the environment. The agents can solve tasks of different complexity level. Some agents are designed to perform simple tasks (easy-task), such as acquiring data from the environment, while other agents can accomplish more complex tasks (hard-task). In order to monitor an event of interest, the preliminary step of the multi-agent system is to allow the agents to accomplish the easy-tasks, collecting the environmental and contextual data necessary to depict the event. Then, agents, capable of accomplishing hard-tasks, are sent to achieve cognitively complex tasks to enrich the knowledge base with ontology data related to the event and situations detected.
The whole process is started by the collector agent in the *Knowledge Collecting Station*, which provides the vehicle agents with the geographical coordinates of an area to inspect (where some events can happen, i.e. fire, high-speed roads, traffic congestion). When the vehicle agents receive the coordinates, they start to monitor the area and acquire data from the environment to return the sensed data to the collector. After a preliminary processing, each agent, targeted at the retrieval of specific data, codes the raw data into semantic knowledge. Each agent can generate high-level knowledge, thus it is enhanced with an ontology model and a semantic inferential engine. The *TrackPOI* ontology model, discussed in Chapter 4, represents scenes set in road, wooded,

urban environments, and it also defines spatio-temporal relations among the detected scene objects. The inferential engine generates new statements by reasoning over the initial facts: consequential concepts and events, modeled as ontological assertions, representing actions and interactions among the scene objects (i.e., walking, grouping, traffic, etc.) are the output of the ontology reasoning.

Summarizing, the data, sensed by the agent vehicles, are transformed into semantic assertions and processed by the inference engine, that elicits new high-level knowledge about that data helping to describe the scenario. In other words, the agent can acquire information, reason about it, and generate a high-level reading of the acquired local information.

6.2.3 A reasoning model for agent UVs: FCM

The agents, in the presented model, can evaluate the scene and find the most suited rescuers by defining and running an FCM. An FCM is a knowledge-driven methodology suited to design complex decision systems, that exploit causal reasoning to make decisions. FCM has been defined By Kosko [177] by synergistically combining neural networks with the fuzzy logic. FCM is a representation of a mental model in terms of concepts, that characterize behaviours and functionalities of a knowledge-based system [178]. FCM can be defined on a specific domain to represent and analyse articulated problems, that can be represented in terms of heterogeneous concepts and eventual causal relations among them. Consequently, the model is perfectly suited to represent knowledge on a complex evolutionary scenario, composed of various features. In fact, the FCM allows the scenario evolution analysis by varying its features and analyse configurations different from the initial ones.

FCMs are designed by experts to deal with real world applications in various contexts, such as, for example, the political field and international relations [179, 180]. These cognitive models have also been widely explored in system control to improve control environment [181], to model actors' intelligence and give better support to decision-making tasks [182], to support failure modes and effect analysis [183], and model system supervisors [184].

An example of FCM is reported in Figure 6.3: the model is represented as a directed graph, where the oval nodes represent the FCM concepts. Generally, a concept can represent a specific entity, a variable or a state of the problem with a specific value, generally taken from the $[0, 1]$ interval. The directed edges, linking the concepts in the model, represent causal relations among couples of concepts. Given the concepts C_1 and C_2 from the FCM shown in figure, the directed edge (C_1, C_2) going from C_1 to C_2 represents the causal impact of C_1 on C_2. The edge sign expresses the way C_1 variations impact on C_2, in other words they cause an increase or decrease of the C_2 value. Therefore, in presence of a positive sign, an increase of the C_1 value causes an increase in the value of C_2. On the contrary, if the edge sign is negative, an increase of the C_1 value triggers a decrease in the value of C_2. The edge value is a fuzzy value representing how much the the concept C_1 impacts on the C_2. Generally, the greater the value, the more C_1 variations affect the C_2 value.
According to the edges, FCM concepts are divided into three types: starting concepts, transition concepts and goal concepts. Concepts that do not have edges directed at them are called starting concepts, and represent the input of the model. The starting concepts can not be affected by other concepts. Concepts, that have edges directed at them, are said to be effect nodes, because these edges represent the causal effect that other concepts have on them. The effect nodes include transition and goal concepts. The transition concepts can be affected directly or indirectly by the starting concepts. The

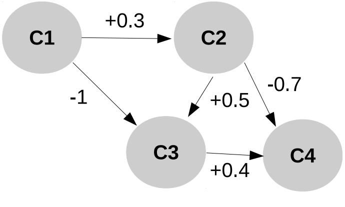


Figure 6.3: An example of FCM

goal concepts do not have edges directed at other concepts, their incident edges are all directed at them. Consequently, the goal concepts can be influenced by the other concepts but they can not affect any other concept. For this reason, goal concept final values represent the output of the FCM.

The agents build the FCM and perform fuzzy causal reasoning through a process called FCM simulation. Once the agents provided initial values for the starting concepts, the FCM simulation or FCM run consists in generating new values for the FCM concepts from the initial concept values and the causal relations among them. The process implements a causal reasoning on the concepts, that, according to the edge weights, allows the FCM concept to activate themselves by assuming new values.

The agents provide the input to the FCM as an initial activation concept vector, containing the initial values for the concepts. After the initial concepts have been initialised or activated, a propagation process in the FCM network is started: each concept updates its value according to its previous value and the values of the concepts connected to it combined with the weights on the edges connecting them. The simulation iteratively propagates the initial activation concept vector until either the map converges to a fixed-point or a maximal number of iterations is reached. Formally, at each iteration t the value C_i^t for the i^{th} concept is calculated by computing the influence of the other concepts C_j^{t-1} on it at previous iteration $t-1$, according to the following formula:

$$C_i^t = f\left(\sum_{j=1}^{n} C_j^{t-1} W_{ji} + C_i^{t-1}\right) \tag{6.1}$$

where $j \neq i$, C_i^t is the updated value for concept i at iteration t, C_j^{t-1} are all the other values of concepts which have a relationship with concept i at iteration k, while W_{ji} is the edge weight between concept i and j. C_i^{t-1} is the value for concept i at iteration $t-1$ and f is a threshold function to squash the result into the $[0, 1]$ interval.

6.2.4 Agent-based knowledge building through FCMs

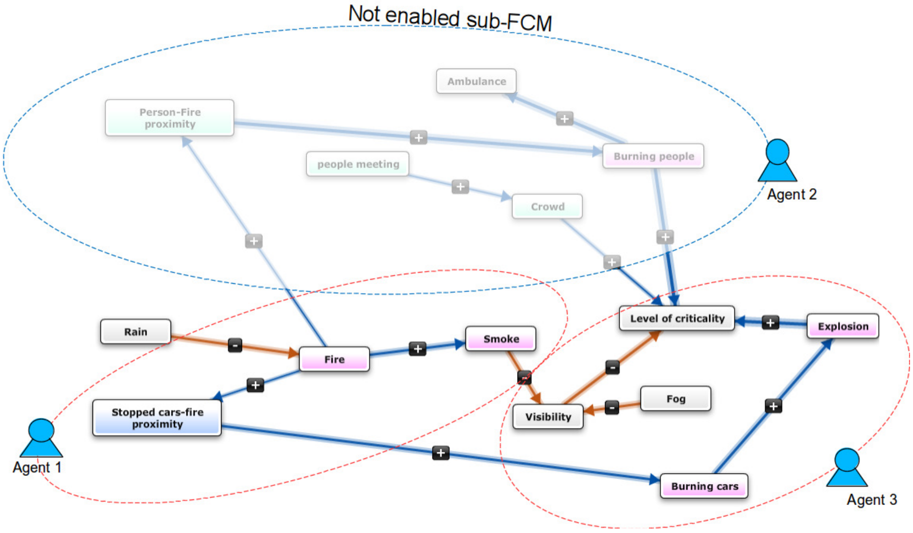


Figure 6.4: FCM building: *Agent 1* (Fire detector) activates *fire* concept, *Agent 3* (Vehicle detector) activates the *Stopped cars-Fire proximity* concept, *Agent 2* (People detector) does not enable the map rooted at *Person-Fire proximity* because it does not find any people in the scenario

Once the vehicle agents, that monitor the area of interest, have gathered information from the environment, they send it to the collector agent. The collector, in the Knowledge Collecting Station, builds a mental representation of the observed scenario dynamically, by combining the individual high-level knowledge produced by the agents. When the agents accomplished their own task, they send the generated data, that generally is in the form of semantic statements, to the collector. Thus, the collector receives the semantic data, that include the concepts obtained by the local agents. These concepts are collected to build the initial knowledge base on the scene, including facts about the scene objects (i.e., people and vehicles) and their interactions with the environment.

As stated in Section 6.2.2, the TrackPOI ontology model is used to represent the collected knowledge on the scene objects and the environment. The ontology also allows the representation of spatio/temporal relations among the scene objects and between the object and the fixed features of the environment, namely Points of Interest (POIs), such as parks, roads, or even banks, stores and others. As seen in Chapter 5, the application of inference to the defined ontology concepts provides new facts expressing events and activities carried out by the scene objects , such as vehicles going off road, vehicles overcoming speed limits, people crossing the road, people meeting, etc.

The high-level knowledge produced by each vehicle agent, through ontology reasoning, is used to dynamically define a complete mental landscape of the scenario. This scenario landscape is made up of high-level concepts and the causal relations among them. In more details, the ontological concepts and events, generated by the agents, enable the activation of a global FCM delineating causal knowledge acquisition and representation related to a specific scenario.

As stated in the previous section, an FCM represents a mental model as a directed graph, where the nodes represent high-level concepts and the edges model the causal relations among the concepts. The concept can assume values, generally in the range $[0, 1]$ or $[-1, 1]$, representing a specific state of the concept. Then, the sensor data and the high-level events detected by the vehicle agents can be used to specialize the initial concept states.

Figure 6.4 shows an example of FCM, that models high-level knowledge on a road scenario as causally related high-level concepts. The FCM comprises concepts representing people and vehicle features (i.e., “people meeting”, “burning cars”), general events (i.e. “fire”, “explosion”) and general features of the environment (“smoke”, “fog”, “visibility”, etc.).

As stated, the FCM can be run to reason over the concepts and their causal relationships. The whole process can be started by providing the initial concept values to the FCM, and then updating concept values according to the edge weights. The process is iterative and comes to an end when convergence is reached. At each iteration, the concept values change defining a specific FCM configuration, representing an evolution from the first FCM configuration. Since the FCM starting concepts are not influenced by other concepts (i.e., they do not have edges directed at them), they need to be initialised to run the FCM. If the input concepts have values different from zero, the FCM is said to be activated by these concepts. Then, the input concept values are propagated to the other concepts in the FCM.

The FCM extract in Figure 6.4 has been designed by domain experts as a composition of different sub-maps or sub-FCMs. Each sub-FCM represents knowledge on a simple scenario, or specific events, associated to the observed environment. Therefore, the sub-FCMs are often related to specific local knowledge about the scene objects and the environment (i.e., people in a storm, fire in a forest, etc.) produced by the vehicle agents.

As stated, each FCM concept is associated with a number expressing its state. Then, concepts in the sub-FCMs, related to a specific agent, can be activated by the agent that provides them with some values. For instance, the agent can provide speed values for the concept “car speed”, or a presence value for the “fire presence” value. Once the initial concepts have been initialised, the causal inference on the concepts can be performed by running the FCM.

According to the concept states or values provided by the agent vehicles, some sub-FCMs can be activated or not. If the returned values do not activate some initial concepts, the sub-FCMs rooted at these concepts, will not be used for the FCM simulation process. In other words, after the vehicle agents have returned their concepts values to the collector, it initializes the FCM accordingly by setting FCM concepts with the returned values. This way, the collector defines the first FCM configuration by activating only the sub-FCMs rooted at the activated concepts. The FCM is then run on this configuration. The output, generated by running the FCM, is the knowledge inferred on the whole scenario by bridging the local knowledge provided by each vehicle agent.

In order to demonstrate this model, let us consider the FCM model shown in Figure 6.4 composed of three sub-FCMs represented by dotted ovals. We assume that the collector agent sends three vehicle agents to monitor the environment, they are: *Agent 1*, *Agent 2*, *Agent 3*. The three agents are , respectively, assigned with specific tasks: Fire detection (*Agent 1*), People detection (*Agent 2*), Vehicle detection (*Agent 3*). Then, each agent is assigned with one of the three sub-FCMs, modelling knowledge on its task.

Once the agents have processed the initial data, they generate concept states that are returned to the collector. Therefore, *Agent 1* states the presence of fire in the scene by returning a non-zero value for the *Fire* concept; *Agent 3* detects stopped cars in the environment and notifies this result by setting *Stopped cars-Fire proximity* concept with a non-zero value. This way, the agents *Agent 1* and *Agent 3* activate the sub-FCMs they are in charge of (marked by red ovals in figure). The *Agent 2*, in charge of people detection, does not find any people in the environment. As a consequence, the agent fixes the *Person-Fire proximity* concept value to 0. This way, the sub-FCM rooted at this concept (marked by the blue oval) is not activated. The final obtained FCM is built as a composition of the agent-enabled sub-FCMs, and initialised with the values provided by the vehicle agents. The so built and initialised FCM can be run to infer events and asses the level of criticality, represented by the values assumed by the goal concepts (i.e., *Explosion*, *Level of criticality*).

6.3 Consensus-based GDM for UV team scenario interpretation

In recent years, UVs have become common-used devices to perform complex tasks. UVs have been used to act as substitute for humans in tasks, that can be risky or hard to perform. UVs found applications in various fields, including military operations, crowd monitoring, fire fighting, breeding and agriculture management [185, 186, 187, 188]. As seen in the previous chapters, sensor-equipped UVs can recognize people, environments, interpret scene object interactions and events by using refined Computer Vision [185] and Artificial intelligence [144] methodologies.

However, to serve the applications introduced above, the use of a single UV can not be enough because of issues about the sensor reliability, weather, methods used and type of environment. Moreover, a single UV provides only one viewpoint on the environment, that can really constraint the scenario interpretation. For instance, the use of a single UGV does not allow to achieve a complete vision of the whole environment. A fleet of UVs of different type (i.e., UAV, UGV, UUV) and equipped with various sensors and technologies can indeed provide a multi-perspective view of the observed scenario. Since each UV has its perspective on the scene, a global complete scenario comprehension is reachable as an agreement among the various UV perspectives.

A team composed of multiple UVs can be seen in Figure 6.5. The team comprises three UVs, including two UAVs and a UGV, devoted to monitor an urban environment. Each UV can observe the scene from its own angle, and, accordingly, produce its own interpretation of situations occurred. Obviously, UVs in the team can have different interpretations of the scene. Therefore, UVs need to find agreement on the scenario comprehension, which can reflects a more realistic scenario description. This problem can be considered as a Group Decision-Making (GDM) problem, where multiple UVs need to reach an agreed collective interpretation of a real-world scenario from their individual distinct views. The application of consensus measures to a GDM problem allows helping experts to find agreement [189, 190]. and choose a collective solution that better satisfies experts' interpretations[191, 192].

The next sections present a consensus GDM approach to allow multi-UV systems to reach an agreed decision on situations that better depict what happened in the scenario observed by the UVs. Given UVs capable of detecting events, the approach enables each

UV to generate preferences on high-level situations through the fuzzy-based aggregation of the events the UV detected.

UVs are considered as experts in a GDM problem, that have to decide which situations are most suited to describe the scenario. Therefore, each UV can express preferences on situations. Then, an agreed collective interpretation of situations is got by applying consensus measures to the UV individual preferences. The consensus application to the UV GDM problem gurantees the UVs to find agreement on the scenario description.

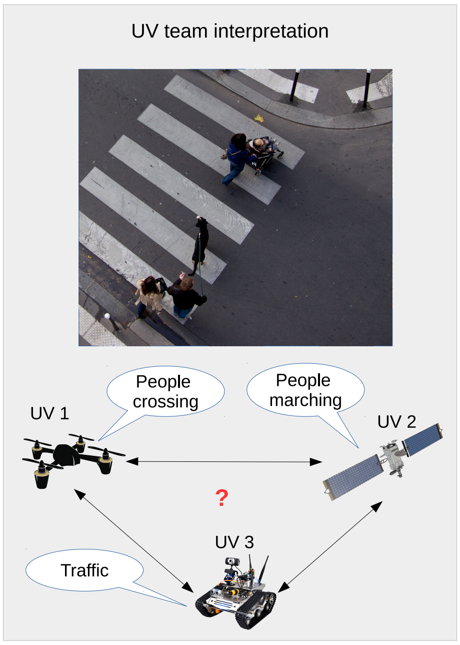


Figure 6.5: Different scene interpretation from the UVs needs to find a collective agreement on the more probable scene description

6.3.1 A consensus GDM model for UVs

The approach, discussed in this section, defines a a consensus-based GDM model to allow a multi-UV system for event detection [144] to reach an agreed interpretation of the monitored scenario.

A logical schema of the approach is shown in Figure 6.6. A UV team, composed of different-type UVs (i.e., UAV, UGV, etc.), monitors an environment and detects events. Each UV is equipped with sensors and enhanced technologies to detect events (*Team of UVs for event detection*). In details, each UV can use the model introduced in Chapter 5 for event and activity detection. As seen, this model applies tracking algorithm to detect scene objects, whose data are coded into semantic statements and fuse with contextual knowledge on the scene by using scene ontologies. Then, ontology reasoning allows the detection events and activities involving the tracked objects.

After UVs have detected events, fuzzy reasoning is applied to allow each UV to recognize situations as fuzzy aggregation of the detected events, and to express preferences for each situation. The *M1* module (*Fuzzy-based UV situation preference generation*) implements a fuzzy modeling of the detected events by using a fuzzy ontology, that allows the UV to define fuzzy linguistic descriptors of the events. The descriptors are then aggregated to get high-level situations, and a preference value for each situation.

The UV preferences are then passed to the *M2* module (*Collective preference and consensus assessment*), that applies the GDM model to lead UVs to decision. The model supports the generation of collective preferences on situations by aggregating the UV individual preferences. Then, consensus and proximity measures are applied to assess the agreement level on situations among the UVs. The modules *M1* and *M2* are detailed in Section 6.3.2 and Section 6.3.3, respectively.

6.3.2 UV preference assessment through fuzzy reasoning

As stated in Chapter 5, the TrackPOI ontology can be used by a smart UV to model contextual knowledge on evolutionary video scenes, and detect high-level events through ontology reasoning. The inferred events detail the actions of the scene objects (i.e., people, vehicles), along with their interactions with the environment

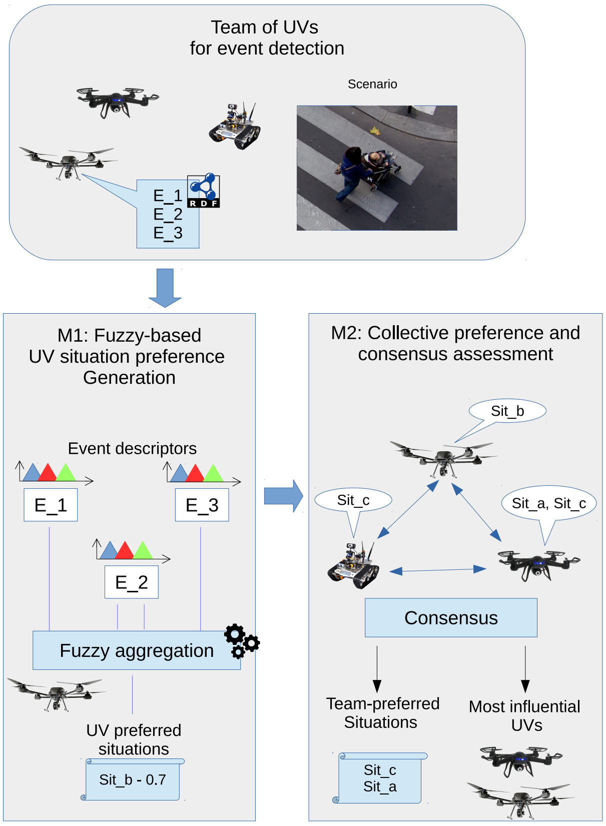


Figure 6.6: Logical view of the model: from UV team event detection to the final team scenario interpretation.

(i.e., POIs) or other scene objects. The events, inferred by each UV, are represented by semantic statements. Therefore, a UV detected event involving the scene object o_1 is represented in triple form as follows: $\langle o_1, e, p\rangle$; where o_1 is a UV detected scene object, e the event kind and p the place where the event occurred.

Some events are more recurrent than others for a place or involve a different number of vehicles and people. Therefore, an analysis of people's (or vehicle) involved in an event, can contribute to explain the event better. To this purpose, given a scene, the *M1* module assesses the frequency of an event kind occurring in a place, according to the number of objects involved in the event.

The event kind frequency is defined as follows.

Event kind frequency. Let us suppose that a UV is monitoring a place p, where the event kind e is happening. A UV detects m scene objects (i.e., people or vehicles) on the place, and recognizes n objects which are involved in the event e. Then, the frequency for the event kind e is calculated as the ratio between n and m.

In order to show how the event frequency is calculated, let us consider the event kind e, occurred in place p, which involves three over six detected people, namely o_1, o_2, o_3). Then, the frequency of e is equal to 0.5. In order to model the event kinds and their frequencies, the TrackPOI ontology is converted to a fuzzy ontology. All the UV-detected event kinds and their frequencies are coded into fuzzy axioms and added to the ontology. These axioms are represented as triples, according to the following format: $\langle u, e, f \rangle$; where u is the instance of a UV, e the event kind and f its frequency value. Therefore, the triple asserts that the UV u detected the event kind e, whose frequency value is equal to f.

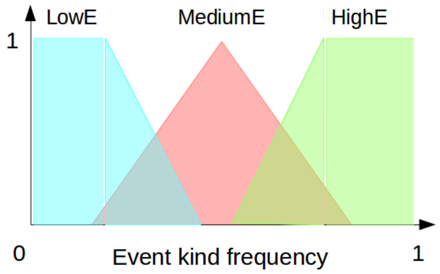


Figure 6.7: Event descriptors for the event e.

According to the event kind frequency values, event descriptors are defined in the fuzzy ontology. The event descriptors are modelled as fuzzy concepts, that describe the event kinds. For instance, three event descriptors of the event kind e are shown in Figure 6.7, they are: *LowE*, *MediumE*, *HighE*. In other words, the event kind e becomes a fuzzy variable that can be associated with three linguistic terms (the three concepts), which are represented by the three fuzzy membership functions shown in the figure. The three concepts represent three different types of people (or vehicles) participating to the event kind e, in a given scene. This way, once the event frequency values have been calculated, the event descriptors describe scene object participation to an event as a fuzzy membership value. For example, if the frequency on the event e is high, the concept *HighE* is more suited to describe scene object involvement in e than *LowE* and *MediumE*.

Since the event descriptors better depict the UV-detected events, they are used to define high-level situation for scene description. A situation is defined as an aggregation of event descriptors, that describe the people or vehicle involvement in the detected events that lead to the situation. Therefore, a situation can be defined, in the ontology, as a fuzzy aggregated concept of event descriptors on distinct event kinds. This situation definition is supported by the fact that the event descriptors represent situational patterns, then, their aggregation leads to describe the overall situation. Given situations defined in terms of the event descriptors, the UV can generate a preference value for each defined situation through maximum concept satisfiability (see Chapter 2, Section 2.4.1). Therefore, running a maximum concept satisfiability query, the UV preference value for a situation can be assessed by aggregating the membership values of event descriptors involved in a situation definition. The aggregation can be computed by using one of the well-known aggregation operators, such as weighted mean, OWA, LOWA, etc. As required by these operators, the event descriptors, involved in the aggregation process for the situation definition, must be chosen. To this purpose, domain experts are in charge of fixing the weights according to how fundamental the event descriptors are in the situation definition. The tasks, accomplished by module $M1$, to allow each UV to define situations and express preferences on them are to be considered as part of a preliminary step (*step 0*). The output of this step (i.e. situations and preferences) are passed to the module $M2$, that applies the consensus-based GDM model to lead UVs to decision. This model is detailed in the next section.

6.3.3 Consensus model

Once each UV expressed preferences on situations, the *M2* module, firstly, allows UVs to build group decision, and then to assess team consensus and detect which UVs lead decision by applying a CRP. Multi-UV system-based situation comprehension is formulated as a GDM problem: the situations are considered as alternatives, on which each UV in the team, seen as an expert, can make evaluations. Then, each UV expresses preferences on the detected situations (Section 6.3.2). Formally, given n UVs and m situations, each UV expresses preferences on the m situations. The preferences expressed by the i^{th} UV are represented as a vector: $P^i = (x_1^i, x_2^i, ..., x_m^i)$, where $P^i \in \mathbb{R}^m, \forall i = 1, 2, ..., n$. The preference x_j^i, in the preference vector P^i represents how much the i^{th} UV prefers the j^{th} situation over the others.

Our UV systems can be composed of different types of UV (ground, aerial, sensor-based, etc.), each one with different features and capabilities. Moreover, weathering such as humidity and luminosity, or other environmental features (i.e., radioactive areas, dense forest), can drop performances of some UVs. For this reason, a reliability degree is associated with each UV, more formally, w_i is the reliability weight associated with the i^{th} UV. Just to give an example, let us consider a UV team composed of 3 UVs (i.e., UV_1, UV_2, UV_3) where UV_1 and UV_3 are equipped with an action camera, and UV_2 with an infra-red camera. The UV weights are fixed, for example, according to how the task assigned to the UV and the environment impact on its performances. If the team has to patrol an area affected by low luminosity, UV_2 will provide more accurate results than the other UVs. Therefore, a reasonable weight assignment to (UV_1, UV_2, UV_3) could be (0.5, 1, 0.5).

Since UV preference vectors represent the information obtained by the single UVs on the scene, the aggregation of these preference vectors contains the overall information of all the UVs that participate in the GDM process. This aggregation provides the collective preferences on situations. Then, consensus and proximity measures are applied to assess group decision reliability. The formal model behind this approach can be summarized in three main steps, described as follows:

1. **Collective preferences.** The model aggregates UV preferences to define a collective preference vector on the situations. The collective preference vector $cp = (cp_1, cp_2, ..., cp_m)$ is composed of m elements, where the j^{th} element (cp_j) represents the team preference on the j^{th} situation. Let $P^i = (x^i_1, x^i_2, ..., x^i_m)$ and w_i be, respectively, the preference vector and the weight associated to the i^{th} UV, cp_j is calculated as the weighted arithmetic mean of the UV preferences on the j^{th} situation:

$$cp_j = \frac{1}{\sum_{k=1}^{n} w_k} \sum_{i=1}^{n} \left(x^i_j \cdot w_i\right) \tag{6.2}$$

   where $j = 1, 2, ..., m$.

   The cp_j value represents the global aggregated preference value on the j^{th} situation. The higher the value, the more the situation is preferred by the team. The collective preference vector (cp) represents the final group decision on each situation.
   After the assessment of the final group decision, our consensus model assesses the general level of agreement among the UVs in the team and which UVs lead the group decision by calculating the consensus (step 2) and proximity measures (step 3), respectively.

2. **Consensus.** The consensus measures take into account UV preferences, which are aggregated to determine different levels of consensus degree among the UVs in the team. As aggregation operator, our model uses the power average mean operator, that has been proven to achieve good performances for decision-making in recent literature [193]. Therefore,

the built consensus process is articulated in three levels of aggregation:

Level 1, Similarity vectors among pairs of UVs. At the first level, consensus among pairs of UVs is determined. In order to detect the similarity on preferences among two UVs, a similarity vector is defined by comparing the two UV preference vectors. Similarity vectors are determined for all the pairs of UVs involved. Therefore, given n UVs, t similarity vectors are assessed, where $t = n \cdot (n-1)/2$. Let P^i and P^j be the preference vectors for the i^{th} and j^{th} UVs, respectively, the similarity vector SV^k among the UV pair is calculated as the distance among the UV preference vectors:

$$SV^k = \mid P^i - P^j \mid \tag{6.3}$$

where $k = 1, 2, ..., t$; $i, j = 1, 2, ..., n$ and $i \neq j$

Level 2, Consensus on situations (cs). The consensus degree among all the UVs on each situation (cs) is got by aggregating the similarity vectors among the UV pairs. Given the similarity vector $SV^k = \left(sv_1^k, sv_2^k, ..., sv_m^k\right)$ among the preference vectors P^i and P^j where $i, j = 1, 2, ..., n$ and $i \neq j$, the consensus degree cs_j among all the UVs on the j^{th} situation is calculated as the power average mean of the j^{th} element in all the similarity vectors:

$$cs_j = \left(\frac{1}{t}\sum_{k=1}^{t} \mid sv_j^k \mid^p\right)^{\frac{1}{p}} \tag{6.4}$$

where $j = 1, 2, ..., m$, $t = n \cdot (n-1)/2$ and p is the p-norm power value.

Consensus on situations degree (cs) identifies which situations the UVs are at odds on, and consequently to judge how the group decision on each situation is reliable.

Level 3, Consensus on the relation (cr). The aggregation of the consensus on situations (cs) provides the consensus among UVs on all the situations (or consensus on the relation) expressed as a unique value. Consensus degree on all the situations (cr) is calculated as the power average mean of the consensus degree on each situation (cs_j):

$$cr = \left(\frac{1}{m} \sum_{j=1}^{m} \mid cs_j \mid^p \right)^{\frac{1}{p}} \tag{6.5}$$

Consensus on the relation (cr) provides a unique cumulative measure to assess the agreement among UVs in the team on all the situations. The closer cr is to 0, the more UVs are in agreement on all the situations and the more reliable is the final group decision (cp). The cr value can be used to accept or discard the group decision on the situations.
In the case study discussed in Section 6.3.4, consensus degrees are calculated by setting $p = 2$ to Equation 6.4 and Equation 6.5.

3. **Proximity measures.** In order to identify which UVs lead the group decision, or which UVs mostly disagree with the group decision about situations, our model defines some proximity measures. These measures are directly calculated on UV preferences and team preference (cp), two levels of proximity measures are built:

   Level 1, Proximity on situations (*ps*). This measure assesses how much the single UV preferences are in agreement with the collective preferences of the team. Given the preference vector P^i for the i^{th} UV and the collective preference vector cp (Equation 6.2), the proximity measure of P^i to the collective preferences is calculated as follows:

$$ps^i = cp - P^i \tag{6.6}$$

   where $i = 1, 2, ..., n$

   The higher ps^i absolute values, the more P^i preferences differ from team preferences. The negative sign of ps^i_j value represents that the i^{th} UV prefers the j^{th} situation more than team. On the contrary, a positive sign means that the UV prefers the j^{th} situation less than team.

Level 2, Cumulative proximity on situations (*cps*). Proximity measures on situations (*ps*) express proximity on each situation. In order to represent the agreement among the single UV and the team on all the situations, cumulative proximity measures have been defined. The cumulative collective preference (*ccp*) is calculated as the arithmetic mean of the elements in the collective preference vector (*cp*):

$$ccp = \frac{1}{m}\sum_{j=1}^{m} cp_j \tag{6.7}$$

Given the preference vector $P^i = (x_1^i, x_2^i, ..., x_m^i)$ for the i^{th} UV, the cumulative preference for the i^{th} UV among all the situations is the arithmetic mean of its elements:

$$cuv^i = \frac{1}{m}\sum_{j=1}^{m} x_j^i \tag{6.8}$$

where $i = 1, 2, ..., n$

The cumulative proximity on all the situations (*cps*) for the i^{th} UV is calculated as the difference between the cumulative collective preference and the cumulative preference for the UV:

$$cps^i = ccp - cuv^i \tag{6.9}$$

where $i = 1, 2, ..., n$

The cps^i value represents how much the decision of the i^{th} UV differs from the final group decision. The higher cps^i absolute value, the more P^i preferences differ, on average, from team preferences on all the situations. Then, UVs, which lead the group decision, will present the lowest cumulative proximity values.
The negative sign of cps^i means that the i^{th} UV expresses

higher preferences, on average, than team on all the situations. The positive sign means the opposite: P^i preferences are averagely lesser than team preferences on all the situations. In essence, the sign of cps^i allows to state if the i^{th} UV is optimist or pessimist on all the situations with respect to the group decision.

The same scenario is taken several times by the UVs by keeping their configurations and positions. This way, UVs acquire several distinct measurements of the same scenario, each one of them is considered as a GDM round. Then, the round with the highest consensus (cr) among the UVs is chosen, and the collective preferences (cp) expressed on situations during this round are considered as the final group decision.

The consensus and proximity degrees are used to annotate axioms on situations and UVs. Accordingly, the system can detect the most plausible situations and UVs leading the group decision by queries.

6.3.4 An application to a real-world scenario

This section presents a case study to show how our model works in a real-world scenario. Let us consider the road scenario shown in Figure 6.8. The scenario involves some people crossing and others walking near the road. Then, let us suppose that a team of six UVs, reached a place, is monitoring the area, where the scene shown in Figure 6.8 is happening. Each UV can detect the five people in the scene through video tracking, other mobile objects are filtered out (i.e., *obj*_6 in the figure). The event detection model, introduced in [122], allows each UV to build knowledge on the tracked objects and the environment by using the TrackPOI ontology. The UV applies reasoning over the built knowledge to detect events as ontology axioms (i.e., subject-predicate-object triples), where the event kind (predicate) is related to the person involved (subject) and the place where the event occurred (object).

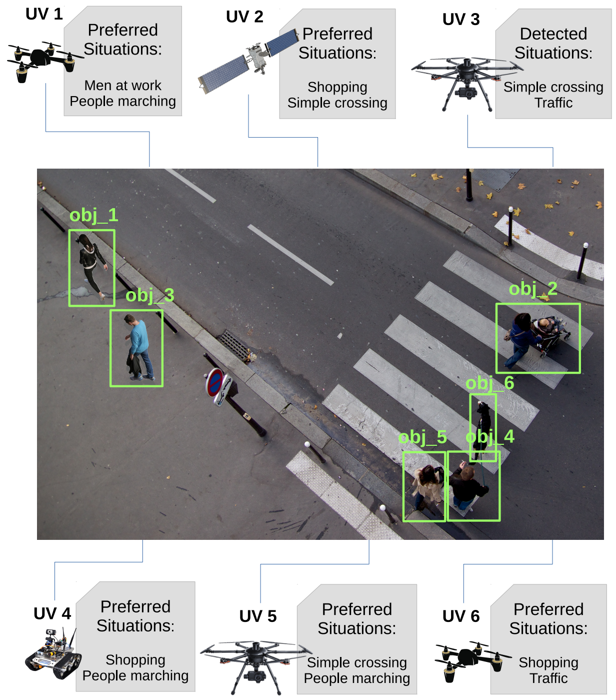


Figure 6.8: Case study: a team of 6 UVs observes and interprets a road scenario.

For instance, the axioms, shown as Turtle[1] code in Listing 6.1, describe the events detected by the *UV_1* involving the detected people and POIs.

```
obj_1 trackpoi:isNear shop .
obj_1 trackpoi:goingTowards building .
obj_2 trackpoi:walkingInside route .
obj_2 trackpoi:crossing route .
obj_3 trackpoi:isNear shop .
obj_4 trackpoi:walkingInside route .
obj_4 trackpoi:crossing route .
obj_5 trackpoi:walkingInside route .
obj_5 trackpoi:crossing route .
```

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

Listing 6.1: Events inferred by *UV_1*: the event is an axiom where the *subject* is the person or vehicle involved, the *predicate* is the kind of the event and the *object* the place where the event occurred.

According to Section 6.3.2, the *M1* module achieves a preliminary step (identified by *0*) targeted at UV preferences setting, then the *M2* module leads UVs to the final group interpretation through other steps (identified by *1-3*).

0. **Situation and preference generation.** The frequencies associated with each event kind detected by the UVs are calculated. For instance, the *crossing* event kind, in Listing 6.1, detected by the UV_1, involves 3 among 5 detected people: *obj*_2, *obj*_4 and *obj*_5 (see lines 4, 7 and 9), then its frequency is 0.6. As Listing 6.2 shows, frequencies support the definition of fuzzy ontology axioms. Precisely, in Listing 6.2 the fuzzy axioms describe the event kinds with their frequencies generated by *UV_1* on the events it detected (see Listing 6.1).

```
(instance UV_1 (= hasGoingTowards 0.2) 1.0 )
(instance UV_1 (= hasCrossing 0.6) 1.0 )
(instance UV_1 (= hasWalkingInsideRoute 0.6) 1.0 )
(instance UV_1 (= hasNearShop 0.4) 1.0 )
(instance UV_1 (= hasVrunning 0) 1.0 )
```

Listing 6.2: Axioms on UVs and events in FuzzyDL syntax: the events and their frequencies are added to the ontology as annotated triples (axioms) which relate the UV (*subject*) with the event kind (*predicate*) and its frequency (*object*). The last value for each triple is the axiom truth degree.

Recall that the event is described by three descriptors: for example, the event kind *GoingTowards* is described by the three event descriptors *LowGoingTowards*, *MediumGoingTowards* and *HighGoingTowards*. Once the event frequencies are assessed, the situation can be revealed and their preference computed for each UV. Let us consider the *People* *marching* situation described in Listing 6.3. It is an aggregation of four event descriptors: $HighGoingTowards$, $HighWalkingInsideRoute$, $LowCrossing$ and $LowVRunning$. The *People marching* situation happens when "many people walking in the same direction" $HighGoingTowards$, "many people walking inside the road area" $HighWalkingInsideRoute$, "few people crossing the road" $LowCrossing$ and the "limited presence of vehicles running on the road" $LowCrossing$. Let us suppose that the event descriptors contribute equally (equal weight, e.g., 0.25) to the situation modeling.

```
(define−concept people_marching (w−sum
(0.25 (some hasGoingTowards HighGoingTowards))
(0.25 (some hasWalkingInsideRoute HighWalkingInsideRoute))
(0.25 (some hasCrossing LowCrossing))
(0.25 (some hasVrunning LowVrunning)) ))
```

Listing 6.3: The definition of people marching situation: the situation is defined as an aggregated fuzzy concept of distinct event descriptors.

At this point, it is possible to compute the preference of UV_1 on the *People marching* situation, by query-based maximum concept satisfiability. In general, the UV preference on a situation is generated by querying the maximum concept satisfiability over the UV instance and its event kind frequencies. Table 6.1 shows the preferences generated by the six UVs on the *People marching* situation. Given the *People marching* concept defined in Listing 6.3, the query is applied to UVs over their frequency values for the 4 event kinds involved in the *People marching* concept (from the second to the fifth column). The last column shows the query results representing the preference values for each UV on the situation. The higher the preference value, the more the UV considers the situation suited to describe the observed scenario. In this example, the *People marching* situation is considered very suited for scenario description by *UV_4*.

1 **Collective preferences.**

Table 6.1: UV preference generation for the *people marching* situation on the event kinds: goingTowards (gt), walkingInsideTheRoute (wir), crossing (crs), vRunning - vehicle running on route (vr).

UV(#)	**Event frequencies**				**Preference**
	gt	**wir**	**crs**	**vr**	**query result**
UV_1	0.20	0.60	0.60	0	0.39
UV_2	0.90	0.75	0.60	0.32	0.56
UV_3	0.50	0.20	1.00	0.82	0.11
UV_4	0.72	0.97	0.11	0.21	0.86
UV_5	0.72	0.11	0.81	0.31	0.30
UV_6	0.12	0.18	0.18	0.71	0.20

Five situations can occur in the scene shown in Figure 6.8: *simple crossing*, *men at work on the road*, *people marching*, *traffic* and *shopping*. UVs generate preferences on these situations, as reported in Table 6.2.

Table 6.2: Preferences of six UVs on five situations: simple crossing (crs), men at work on the road (wrk), people marching (mar), traffic (trf), shopping (sho).

UV(#)	**Preferences on situations**				
	crs	**wrk**	**mar**	**trf**	**sho**
UV_1	0.29	0.75	0.39	0.00	0.29
UV_2	0.79	0.61	0.56	0.04	0.99
UV_3	0.71	0.00	0.11	0.46	0.00
UV_4	0.37	0.65	0.86	0.25	0.71
UV_5	0.81	0.12	0.30	0.26	0.00
UV_6	0.00	0.00	0.20	0.59	0.77

According to Equation (6.2), the team collective preference vector (*cp*) is calculated, its values are reported in Table 6.3. The team prefers *simple crossing (crs)* as the most suited situation to describe the observed scenario, while *traffic (trf)*

Table 6.3: The collective preferences: the values represent the group decision on each situation.

Measure	**Collective preferences on situations**				
	crs	**wrk**	**mar**	**trf**	**sho**
cp	0.37	0.27	0.30	0.20	0.35

and *men at work on the road (wrk)* are the least eligible situations to depict what happened. This result represents the final group decision. Since the considered scenario does not present any condition that can compromise performances of any UV, for sake of simplicity, we assumed each UV as equally reliable by assigning their weights to 1 in Equation (6.2).

2 **Consensus.** Once the collective preferences have been generated, the consensus measures, described in Section 6.3.3, allow evaluating the agreement level among the UVs. Recall that our consensus model is composed of three levels of aggregation, as described in Section 6.3.3. The *Level 1, Similarity vectors among pairs of UVs* assesses the similarity among pairs of UVs. Similarity vectors are the rows in Table 6.4, and represent how UV pairs agree about situations. For example, in the case of the *simple crossing* situation, the UV pairs, which agree most, are the couples: (*UV_1*, *UV_4*), (*UV_2*, *UV_3*), (*UV_2*, *UV_5*) and (*UV_3*, *UV_5*). The aggregation of the similarity vectors on the UVs, according to *cs* measure (Equation 6.4), allows the evaluation of the consensus degree among the UVs on each situation. The results are represented as the *cs* vector and reported in Table 6.5. Let us notice that the team agrees mostly on the *traffic (trf)* situation while strongly disagrees on *shopping (sho)* situation.

Table 6.4: Similarity among UV pairs are assessed on the situations: simple crossing (crs), men at work on the road (wrk), people marching (mar), traffic (trf), shopping (sho).

UV(#)	**Similarity among UV pairs on situations**				
	crs	**wrk**	**mar**	**trf**	**sho**
UV_1 - UV_2	0.50	0.14	0.17	0.04	0.71
UV_1 - UV_3	0.43	0.75	0.29	0.46	0.29
UV_1 - UV_4	0.09	0.10	0.47	0.26	0.43
UV_1 - UV_5	0.52	0.63	0.09	0.26	0.29
UV_1 - UV_6	0.29	0.75	0.19	0.59	0.49
UV_2 - UV_3	0.07	0.61	0.45	0.42	0.99
UV_2 - UV_4	0.41	0.03	0.30	0.20	0.29
UV_2 - UV_5	0.02	0.49	0.25	0.22	0.99
UV_2 - UV_6	0.79	0.61	0.36	0.55	0.23
UV_3 - UV_4	0.34	0.65	0.75	0.21	0.71
UV_3 - UV_5	0.09	0.12	0.19	0.20	0.0
UV_3 - UV_6	0.71	0.0	0.09	0.14	0.77
UV_4 - UV_5	0.44	0.53	0.56	0.01	0.71
UV_4 - UV_6	0.37	0.65	0.66	0.35	0.06
UV_5 - UV_6	0.81	0.12	0.10	0.34	0.77

Table 6.5: The consensus degree among the UVs on each situation (*cs* vector).

Measure	**Consensus on situations**				
	crs	**wrk**	**mar**	**trf**	**sho**
cs	0.46	0.50	0.39	0.33	0.60

Starting from the *cs* vector, the consensus on the relation (cr) measure is calculated (Equation 6.5). Its value is 0.46, which means that, on average, the UVs agree on all situations at 54%. In other words, they partially agree on all situations.

3 **Proximity measures.** To detect the UVs leading the group decision, the proximity *ps* (Equation 6.6) of each single UV to team is assessed. The resulting *ps* vectors are shown in Table 6.6. The values on the i^{th} row show how the i^{th} UV preferences differ from the team preferences on each situation. This measure detects which situations the single UVs are most at the odds on with the team, as well as which UVs lead the group decision on each situation. For example, *UV_2* and *UV_5* most disagree about the team preference on simple crossing (crs) situation, whereas the *UV_4* and *UV_1* lead the decision process on this situation. The *people marching (mar)* situation is the one with the greatest number of decision leaders (i.e. *UV_1*, *UV_5* and *UV_6*).
To detect UVs, who lead the decision process on all situations, the *cumulative proximity on situations* measure is employed (Equation 6.9). Table 6.7 shows *cps* vector, let us notice that *UV_5* leads the group decision on all the situations, while *UV_2* and *UV_4* present the most different decisions from the final group decision.

6.4 Discussion

The main novelties evidenced in the presented approach are basically related to team-based activities, as listed as follows.

- **Team decision evaluation through a consensus process:** the main trends in literature propose team solutions for target searching [194], path planning [195] and team control [185] by collecting and fusing UV information. These methods do not evaluate how reliable the final outcome of the task is. Our model, instead, provides the agreement

Table 6.6: Individual UV proximity to team on the five situations: simple crossing (crs), men at work on the road (wrk), people marching (mar), traffic (trf), shopping (sho).

UV(#)	**UV-team proximity on situations**				
	crs	**wrk**	**mar**	**trf**	**sho**
UV_1	0.08	-0.48	-0.09	0.20	0.06
UV_2	-0.42	-0.35	-0.26	0.16	-0.65
UV_3	-0.34	0.27	0.20	-0.26	0.35
UV_4	-0.0008	-0.38	-0.56	-0.05	-0.37
UV_5	-0.44	0.15	-0.001	-0.06	0.35
UV_6	0.37	0.27	0.10	-0.39	-0.43

Table 6.7: UV cumulative proximity to team: each row shows how the single UV decision differs from group decision on all the situations.

UV(#)	**Cumulative UV-team proximity (cps)**
UV_1	- 0.05
UV_2	-0.30
UV_3	0.04
UV_4	-0.27
UV_5	-0.0005
UV_6	-0.02

level assessment among the UVs in the team. The reached agreement states how reliable the team solution is and how suitable the individual evaluations of the UVs in the team are. Therefore, the *cr* measure also provides the reliability degree of the final group decision.

- **Scene comprehension and situation detection:** to reach the situation awareness, the design of approaches for target searching [196], and event detection [185, 144] requires to analyze data on the observed environment and the detection and monitoring of events and possible situations [144]. High level abstraction is not easy to achieve, because UVs can recognize target, accomplish tasks (firefighting [186], crowd monitoring [185], etc.), but emulating the human capability of understanding and synthesis when observing a scenario is also a challenge. Therefore, it would be interesting to assess the extent to which the multi-UV system correctly detected a specific situation. Our model determines the situations, which better describe the scene, by a collective consensus reached by the UV team on the scene. Thanks to consensus degree, it is possible to check how reliable is the system evaluation on a specific situation.

- **Checking when to re-plan UV missions:** many approaches focus on processing UV features and positions to handle the cooperation [185, 195] and apply decision-making methods to allow UVs to decide when to patrol an area [197]. An important feature is the UV reliability in scenario detection, especially if compared with the remaining team, to take possible replanning individual target mission into account. If, for example, the UVs do not find agreement on the detected situations, they need to re-plan their missions to acquire better information.

- **Individual vs. team perspective on scene interpretation:** the main trends in literature employ information sharing among the UVs to build a common global knowledge on the observed scene [194, 196, 185]. Our model, instead, uses proximity measures between each UV and the team, to check how each UV perspective on the scene meets the final collective outcome. This measure describes the extent to which the scene comprehension takes into account evaluations from different perspectives.

- **Automatic critical UVs detection:** UV positions as well as specific sensor-based features are often used to UV control and path planning [198, 195] , especially in team mission. For instance, team reliability can be guaranteed by knowing if a UV has damaged or unreliable sensors, or other kinds of issues. Our model is designed to evaluate how much the single UV agrees with its team. This way, UVs that do not reflect the team behavior (measured by the proximity measure) need to be fixed.

Although the model has several benefits, the proposed approach could suffer from some drawbacks:

- **Complex scenario interpretation issues:** the model has been demonstrated on straightforward case studies. Crowded scenario or scenes populated with numerous, heterogeneous targets could cause problems in object detection and tracking [185], with consequent effects on the event and situation identification. These issues could affect the effectiveness of the consensus calculation, based on the semantic reasoning performance for the situation identification.
- **UV teams have to work on the situations from the same scenario:** GDM with consensus modeling enables experts to express a judgment/preference on the prefixed set of possible alternatives [189]. Our model works with UVs devoted to a predefined set of possible situations. [144]. In these systems, indeed, there is the need to find some common aspects on which the UVs can interact to improve scenario comprehension.

Chapter 7

Conclusion

As aforementioned, the contributions of this thesis are manifold, as reported on the following.

- **A geometrical structure for concept learning.** The first aim was to explore knowledge extraction from structured and unstructured data to support smart surveillance applications with Unmanned Vehicles (UVs). Therefore, a layered geometrical structure, namely the simplicial complex, has been introduced to extract high-level concepts from texts.

- **Ontology modeling of dynamic scenarios.** Subsequently, the focus has been set on knowledge extraction from multimedia files generated by UVs. The main thesis contribution to this problem is an ontology-based approach that allows high-level knowledge modelling of the scene as composed of mobile actors, detected through video tracking, and fixed environmental entities. Then, ontology reasoning has been explored to generate new knowledge on the scene.

- **A human-like event detection model.** The interpretation of an evolutionary scenario requires not only recognition of scene actors (i.e., people, vehicles) but also the interpretation of events and activities. To this purpose, this dissertation investigated methods to understand the scene at various levels of detail (i.e., scene actors, events, activities and situations). The main contributions on these topics include knowledge-based frameworks, that based on the introduced ontology-based approach, incrementally build knowledge on tracking data to depict activities and situations. The proposed frameworks allow knowledge abstraction to accomplish human-like descriptions of the overall monitored scene.

- **An agent-based modeling for cooperative devices to scene knowledge building.** The last contribution of this dissertation concerns models to let systems, composed of multiple UVs and sensors, achieve scenario comprehension. The main issues related to those systems is knowledge sharing and combination to achieve group interpretation of a scenario. This thesis proposed an agent-based modelling of the devices to let them build knowledge on the scene cooperatively.
- **A consensus-based GDM model for UV-group-agreed scenario interpretation.** Multiple smart devices can perform multi-view scene monitoring, however, they can provide different interpretations of the scenario. This thesis contributes to tackle this problem by introducing a new consensus-based Group Decision Making (GDM) approach to let devices achieve an agreed team interpretation of the scene.

Case studies and experimentations, presented throughout the thesis, demonstrated the applicability of the proposed models for enhancing UV systems to support humans in complex surveillance and monitoring tasks.

Despite the benefits introduced by the presented models, they may suffer from some limitations, that are reported on the following.

- **The projection of the current environment state (level 3 SA) is not fully supported**. According to Endsley definition, as reported in Section 2.2, the Situation Awareness (SA) is defined as the perception of the environmental elements (level 1 SA), the comprehension of the current environment (level 2 SA) and the projection of the current environmental state in the future (level 3 SA). The ontology-based models support the comprehension of the current state of the scene (level 1 SA and level 2 SA), but they can not make predictions on future evolutions of the scene and fully satisfy the final level of Situation Awareness (level 3 SA).

- **GDM requires UVs returning the same output.** The use of the GDM for UV team scene interpretation allows a group of devices to reach an agreed multi-view interpretation of the scene. However, the GDM solution requires devices capable of making judgements on the same set of alternatives.

Future works will focus on addressing the limitations to the presented models, and explore solutions to evaluate the extent to which the knowledge-based models presented reflect a human-like behaviour, as discussed in Section 5.3.3. This evaluation will have to support the implementation and use of the proposed solutions in various application contexts.

Bibliography

[1] N. D. Rodríguez, M. P. Cuéllar, J. Lilius, and M. D. Calvo-Flores, "A fuzzy ontology for semantic modelling and recognition of human behaviour," *Knowledge-Based Systems*, vol. 66, pp. 46 – 60, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705114001385

[2] J. Gómez-Romero, M. A. Patricio, J. García, and J. M. Molina, "Ontology-based context representation and reasoning for object tracking and scene interpretation in video," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7494 – 7510, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417410014818

[3] L. Snidaro, J. García, and J. Llinas, "Context-based information fusion: A survey and discussion," *Information Fusion*, vol. 25, pp. 16 – 31, 2015.

[4] C. F. Crispim-Junior, V. Buso, K. Avgerinakis, G. Meditskos, A. Briassouli, J. Benois-Pineau, I. Y. Kompatsiaris, and F. Bremond, "Semantic event fusion of different visual modality concepts for activity recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1598–1611, Aug 2016.

[5] X. Wang, H. Song, and H. Cui, "Pedestrian abnormal event detection based on multi-feature fusion in traffic video," *Optik - International Journal for Light and Electron Optics*, vol. 154, pp. 22 – 32, 2018.

[6] S. Growe and J. Bückner, "Knowledge based interpretation of remote sensing images using semantic nets," *Photogrammetric Engineering and Remote Sensing*, vol. 65, 07 1999.

[7] M. Endsley, "Endsley, m.r.: Toward a theory of situation awareness in dynamic systems. human factors journal 37(1), 32-64," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, pp. 32–64, 03 1995.

[8] J. Dong, G. Wu, T. Yang, and Z. Jiang, "Battlefield situation awareness and networking based on agent distributed computing," *Physical Communication*, vol. 33, pp. 178 – 186, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187449071830452X

[9] T. Berners-Lee and M. Fischetti, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*, 1st ed. Harper San Francisco, 1999.

[10] T. BERNERS-LEE, J. HENDLER, and O. LASSILA, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001. [Online]. Available: http://www.jstor.org/stable/26059207

[11] E. Sanchez, *Fuzzy Logic and the Semantic Web (Capturing Intelligence)*. New York, NY, USA: Elsevier Science Inc., 2006.

[12] S. Calegari and D. Ciucci, "Fuzzy ontology, fuzzy description logics and fuzzy-owl," in *Applications of Fuzzy Sets Theory*, F. Masulli, S. Mitra, and G. Pasi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 118–126.

[13] F. Bobillo and U. Straccia, "The fuzzy ontology reasoner fuzzyDL," *Knowledge-Based Systems*, vol. 95, pp. 12 – 34, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705115004621

[14] J. Morente-Molinera, G. Kou, C. Pang, F. Cabrerizo, and E. Herrera-Viedma, "An automatic procedure to create fuzzy ontologies from users´ opinions using sentiment analysis procedures and multi-granular fuzzy linguistic modelling methods," *Information Sciences*, vol. 476, pp. 222 – 238, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002002551830820X

[15] J. Morente-Molinera, R. Wikström, E. Herrera-Viedma, and C. Carlsson, "A linguistic mobile decision support system based on fuzzy ontology to facilitate knowledge mobilization," *Decision Support Systems*, vol. 81, pp. 66 – 75, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923615001785

[16] S. Calegari and E. Sanchez, "Object-fuzzy concept network: An enrichment of ontologies in semantic information retrieval," *Journal of the American Society for Information Science and Technology*, vol. 59, no. 13, pp. 2171–2185, 2008. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20945

[17] S. Dasiopoulou, V. Tzouvaras, I. Kompatsiaris, and M. G. Strintzis, "Enquiring mpeg-7 based multimedia ontologies," *Multimedia Tools and Applications*, vol. 46, no. 2, pp. 331–370, Jan 2010. [Online]. Available: https://doi.org/10.1007/s11042-009-0387-4

[18] R.-C. Chen, C.-T. Bau, and C.-J. Yeh, "Merging domain ontologies based on the wordnet system and fuzzy formal concept analysis techniques," *Applied Soft Computing*, vol. 11, no. 2, pp. 1908 – 1923, 2011, the Impact of Soft Computing for the Progress of Artificial Intelligence. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494610001432

[19] F. Ali, S. R. Islam, D. Kwak, P. Khan, N. Ullah, S. jo Yoo, and K. Kwak, "Type-2 fuzzy ontology–aided recommendation systems for iot-based healthcare," *Computer Communications*, vol. 119, pp. 138 – 155, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366417310587

[20] E. Herrera-Viedma, L. Martinez, F. Mata, and F. Chiclana, "A consensus support system model for group decision-making problems with multigranular linguistic preference relations," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 5, pp. 644–658, Oct 2005.

[21] H. Wang, Z. Xu, X. Zeng, and H. Liao, "Consistency measures of linguistic preference relations with hedges," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 2, pp. 372–386, Feb 2019.

[22] C.-C. Li, Y. Dong, Y. Xu, F. Chiclana, E. Herrera-Viedma, and F. Herrera, "An overview on managing additive consistency of reciprocal preference relations for consistency-driven decision making and fusion: Taxonomy and future directions," *Information Fusion*, vol. 52, pp. 143 – 156, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253518307462

[23] C. Zuheros, C.-C. Li, F. J. Cabrerizo, Y. Dong, E. Herrera-Viedma, and F. Herrera, "Computing with words: Revisiting the qualitative scale," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 26, no. Suppl. 2, pp. 127–143, 2018. [Online]. Available: https://doi.org/10.1142/S0218488518400147

[24] S. Wan, F. Wang, and J. Dong, "A group decision-making method considering both the group consensus and multiplicative consistency of interval-valued intuitionistic fuzzy preference relations," *Information Sciences*, vol. 466, pp. 109 – 128, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025516308581

[25] M. S. A. Khan, S. Abdullah, A. Ali, and F. Amin, "Pythagorean fuzzy prioritized aggregation operators and their application to multi-attribute group decision making," *Granular Computing*, 04 2018.

[26] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 183–190, Jan 1988.

[27] F. Herrera, E. Herrera-Viedma, and J. Verdegay, "Direct approach processes in group decision making using linguistic owa operators," *Fuzzy Sets and Systems*, vol. 79, no. 2, pp. 175 – 190, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016501149500162X

[28] J. Morente-Molinera, G. Kou, K. Samuylov, R. Ureña, and E. Herrera-Viedma, "Carrying out consensual group decision making processes under social networks using sentiment analysis over comparative expressions," *Knowledge-Based Systems*, vol. 165, pp. 335 – 345, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705118305938

[29] H. Zhang, Y. Dong, F. Chiclana, and S. Yu, "Consensus efficiency in group decision making: A comprehensive comparative study and its optimal design," *European Journal of Operational Research*, vol. 275, no. 2, pp. 580 – 598, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221718309937

[30] C. Li, R. M. Rodríguez, L. Martínez, Y. Dong, and F. Herrera, "Consensus building with individual consistency control in group decision making," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 2, pp. 319–332, Feb 2019.

[31] F. Chiclana, E. Herrera-Viedma, F. Herrera, and S. Alonso, "Some induced ordered weighted averaging operators and their use for solving group decision-making problems based on fuzzy preference relations," *European Journal of Operational Research*, vol. 182, no. 1, pp. 383 – 399, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221706008095

[32] V. Loia, W. Pedrycz, and S. Senatore, "Semantic web content analysis: A study in proximity-based collaborative clustering," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 6, pp. 1294–1312, Dec 2007.

[33] M. Phillips, *Aspects of text structure: An investigation of the lexical organization of text.* Elsevier, 1985.

[34] H. Schütze, "Automatic word sense discrimination," *Comput. Linguist.*, vol. 24, no. 1, pp. 97–123, Mar. 1998. [Online]. Available: http://dl.acm.org/citation.cfm?id=972719.972724

[35] M. Sanderson and B. Croft, "Deriving concept hierarchies from text," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99. New York, NY, USA: ACM, 1999, pp. 206–213. [Online]. Available: http://doi.acm.org/10.1145/312624.312679

[36] V. Loia, W. Pedrycz, and S. Senatore, "P-fcm: a proximity-based fuzzy clustering for user-centered web applications," *International Journal of Approximate Reasoning*, vol. 34, no. 2, pp. 121 – 144, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888613X03000884

[37] R. Y. k. Lau, J. X. Hao, M. Tang, and X. Zhou, "Towards context-sensitive domain ontology extraction," in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, Jan 2007, pp. 60–60.

[38] U. Kruschwitz, "An adaptable search system for collections of partially structured documents," *IEEE Intelligent Systems*, vol. 18, no. 4, pp. 44–52, Jul 2003.

[39] S.-L. Chuang and L.-F. Chien, "Taxonomy generation for text segments: A practical web-based approach," *ACM Trans. Inf. Syst.*, vol. 23, no. 4, pp. 363–396, Oct. 2005. [Online]. Available: http://doi.acm.org/10.1145/1095872.1095873

[40] R. Baeza-Yates, "Graphs from search engine queries," in *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science*, ser. SOFSEM '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-69507-3_1

[41] R. Girju, A. Badulescu, and D. Moldovan, "Automatic discovery of part-whole relations," *Comput. Linguist.*, vol. 32, no. 1, pp. 83–135, Mar. 2006. [Online]. Available: http://dx.doi.org/10.1162/coli.2006.32.1.83

[42] R. Snow, D. Jurafsky, and A. Y. Ng, "Semantic taxonomy induction from heterogenous evidence," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ser. ACL-44. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 801–808. [Online]. Available: http://dx.doi.org/10.3115/1220175.1220276

[43] F. Reichartz, H. Korte, and G. Paass, "Composite kernels for relation extraction," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ser. ACLShort '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 365–368. [Online]. Available: http://dl.acm.org/citation.cfm?id=1667583.1667696

[44] C. Giuliano, A. Lavelli, D. Pighin, and L. Romano, “Fbk-irst: Kernel methods for semantic relation extraction,” in *Proceedings of the 4th International Workshop on Semantic Evaluations*, ser. SemEval ’07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 141–144. [Online]. Available: http://dl.acm.org/citation.cfm?id=1621474.1621502

[45] T. H. Cao, H. T. Do, D. T. Hong, and T. T. Quan, “Fuzzy named entity-based document clustering,” in *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, June 2008, pp. 2028–2034.

[46] I. Diaz-Valenzuela, V. Loia, M. J. Martin-Bautista, S. Senatore, and M. A. Vila, “Automatic constraints generation for semisupervised clustering: experiences with documents classification,” *Soft Computing*, vol. 20, no. 6, pp. 2329–2339, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00500-015-1643-3

[47] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, “Distant supervision for relation extraction without labeled data,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ser. ACL ’09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 1003–1011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1690219.1690287

[48] V. Loia, W. Pedrycz, S. Senatore, and M. I. Sessa, “Web navigation support by means of proximity-driven assistant agents,” *JASIST*, vol. 57, pp. 515–527, 2006.

[49] P. D. Rocca, S. Senatore, and V. Loia, “A semantic-grained perspective of latent knowledge modeling,” *Information Fusion*, vol. 36, pp. 52 – 67, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253516301257

[50] P. Cimiano, A. Hotho, and S. Staab, “Learning concept hierarchies from text corpora using formal concept analysis,” *J. Artif. Int. Res.*, vol. 24, no. 1, pp. 305–339, Aug. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622519.1622528

[51] C. D. Maio, G. Fenza, V. Loia, and S. Senatore, "Hierarchical web resources retrieval by exploiting fuzzy formal concept analysis," *Information Processing & Management*, vol. 48, no. 3, pp. 399 – 418, 2012, soft Approaches to {IA} on the Web. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457311000458

[52] V. Loia and S. Senatore, "A fuzzy-oriented sentic analysis to capture the human emotion in web-based content," *Knowledge-Based Systems*, vol. 58, pp. 75 – 85, 2014, intelligent Decision Support Making Tools and Techniques: {IDSMT}. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705113003110

[53] R. Navigli, P. Velardi, A. Cucchiarelli, and F. Neri, "Quantitative and qualitative evaluation of the ontolearn ontology learning system," in *Proceedings of the 20th International Conference on Computational Linguistics*, ser. COLING '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004. [Online]. Available: https://doi.org/10.3115/1220355.1220505

[54] R. Navigli and P. Velardi, *Ontology Enrichment Through Automatic Semantic Annotation of On-Line Glossaries*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 126–140. [Online]. Available: http://dx.doi.org/10.1007/11891451_14

[55] A. G. Valarakos, G. Paliouras, V. Karkaletsis, and G. Vouros, *Enhancing Ontological Knowledge Through Ontology Population and Enrichment*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 144–156. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30202-5_10

[56] H. Alani, S. Kim, D. E. Millard, M. J. Weal, W. Hall, P. H. Lewis, and N. Shadbolt, "Automatic ontology-based knowledge extraction from web documents," *IEEE Intelligent Systems*, vol. 18, pp. 14–21, 2003.

[57] S. Shehata, F. Karray, and M. S. Kamel, "An efficient concept-based retrieval model for enhancing text retrieval quality," *Knowledge and Information Systems*, vol. 35, no. 2, pp.

411–434, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10115-012-0504-y

[58] E. Agirre, O. Ansa, E. Hovy, and D. Martínez, "Enriching very large ontologies using the www," in *Proceedings of the First International Conference on Ontology Learning - Volume 31*, ser. OL'00. Aachen, Germany, Germany: CEUR-WS.org, 2000, pp. 25–30. [Online]. Available: http://dl.acm.org/citation.cfm?id=3053703.3053709

[59] J. L. Fagan, "The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval," *Journal of the American Society for Information Science*, vol. 40, no. 2, p. 115, Mar 01 1989, last updated - 2013-02-24. [Online]. Available: https://search.proquest.com/docview/1301251887?accountid=14734

[60] E. Cambria and B. White, "Jumping nlp curves: A review of natural language processing research [review article]," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, May 2014.

[61] C. L. Yang, N. Benjamasutin, and Y. H. Chen-Burger, "Mining hidden concepts: Using short text clustering and wikipedia knowledge," in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, May 2014, pp. 675–680.

[62] K. Aas and L. Eikvil, "Text categorisation: A survey." 1999.

[63] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975. [Online]. Available: http://doi.acm.org/10.1145/361219.361220

[64] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval.* New York, NY, USA: McGraw-Hill, Inc., 1986.

[65] A. Kalogeratos and A. Likas, "Text document clustering using global term context vectors," *Knowledge and Information Systems*, vol. 31, no. 3, pp. 455–474, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10115-011-0412-6

[66] A. Tombros and C. J. van Rijsbergen, "Query-sensitive similarity measures for information retrieval," *Knowl. Inf. Syst.*, vol. 6, no. 5, pp. 617–642, Sep. 2004. [Online]. Available: http://dx.doi.org/10.1007/s10115-003-0115-8

[67] X. Liu, J. J. Webster, and C. Kit, "An extractive text summarizer based on significant words," in *Proceedings of the 22Nd International Conference on Computer Processing of Oriental Languages. Language Technology for the Knowledge-based Economy*, ser. ICCPOL '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 168–178. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-00831-3_16

[68] G. Salton and C. Buckley, "Term-weighting Approaches in Automatic Text Retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, Aug. 1988. [Online]. Available: http://dx.doi.org/10.1016/0306-4573(88)90021-0

[69] C. D. Manning and H. Schutze, *Foundations of statistical natural language processing.* Cambridge: MIT press, 1999, vol. 999.

[70] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 371–380, June 2003.

[71] A. Symington, S. Waharte, S. Julier, and N. Trigoni, "Probabilistic target detection by camera-equipped UAVs," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 4076–4081.

[72] M. Ugliano, L. Bianchi, A. Bottino, and W. Allasia, "Automatically detecting changes and anomalies in unmanned aerial vehicle images," in *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2015 IEEE 1st International Forum on*, Sept 2015, pp. 484–489.

[73] M. Siam, R. ElSayed, and M. ElHelw, "On-board multiple target detection and tracking on camera-equipped aerial vehicles," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, Dec 2012, pp. 2399–2405.

[74] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, June 2017.

[75] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, Jan 2017.

[76] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid, "Unsupervised object discovery and tracking in video collections," in *Proceedings of the IEEE international conference on Computer Vision*, 2015, pp. 3173–3181.

[77] F. D. Smedt, D. Hulens, and T. Goedeme, "On-board real-time tracking of pedestrians on a UAV," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on*, June 2015, pp. 1–8.

[78] S. Minaeian, J. Liu, and Y. J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 1005–1016, July 2016.

[79] 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 *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, July 2013, pp. 1–6.

[80] J. T. Lee, C. C. Chen, and J. K. Aggarwal, "Recognizing human-vehicle interactions from aerial video without training," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, June 2011, pp. 53–60.

[81] J. Wang, J. Qian, and R. Ma, "Urban road information extraction from high resolution remotely sensed image based on semantic model," in *21st International Conference on Geoinformatics,*, June 2013, pp. 1–5.

[82] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, “Efficient road detection and tracking for unmanned aerial vehicle,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 297–309, Feb 2015.

[83] V. Bruni and D. Vitulano, “An improvement of kernel-based object tracking based on human perception,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 11, pp. 1474–1485, Nov 2014.

[84] T. Chen and S. Lu, “Object-level motion detection from moving cameras,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2016.

[85] P. Huang, J. Cai, Z. Meng, Z. Hu, and D. Wang, “Novel method of monocular real-time feature point tracking for tethered space robots,” *Journal of Aerospace Engineering*, vol. 27, no. 6, p. 04014039, 2013.

[86] L. Chen, P. Huang, J. Cai, Z. Meng, and Z. Liu, “A non-cooperative target grasping position prediction model for tethered space robot,” *Aerospace Science and Technology*, vol. 58, no. Supplement C, pp. 571 – 581, 2016.

[87] 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. 32 152–32 167, 2015.

[88] M. Piccardi, “Background subtraction techniques: a review,” in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, Oct 2004, pp. 3099–3104 vol.4.

[89] R. Brehar, C. Fortuna, S. Bota, D. Mladenic, and S. Nedevschi, “Spatio-temporal reasoning for traffic scene understanding,” in *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*, Aug 2011, pp. 377–384.

[90] 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 Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1–11, 2016.

[91] M. Eldib, N. B. Bo, F. Deboeverie, J. Nino, J. Guan, S. V. de Velde, H. Steendam, H. Aghajan, and W. Philips, “A low resolution multi-camera system for person tracking,” in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 378–382.

[92] X. Li and H. Lu, “Object tracking based on local learning,” in *2012 19th IEEE International Conference on Image Processing*, Sept 2012, pp. 413–416.

[93] W. Jiang, C. Xiao, H. Jin, S. Zhu, and Z. Lu, “"vehicle tracking with non-overlapping views for multi-camera surveillance system",” in *High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing, 2013 IEEE 10th International Conference on*, Nov 2013, pp. 1213–1220.

[94] Z. Zhang and F. Cohen, “3d pedestrian tracking based on overhead cameras,” in *Distributed Smart Cameras (ICDSC), 2013 Seventh International Conference on*, Oct 2013, pp. 1–6.

[95] Y. Yuan, L. Mou, and X. Lu, “Scene recognition by manifold regularized deep learning architecture,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2222–2233, Oct 2015.

[96] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, “Learning deep features for scene recognition using places database,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 487–495.

[97] Z. Li and J. Ding, “Ground moving target tracking control system design for uav surveillance,” in *Automation and Logistics, 2007 IEEE International Conference on*, Aug 2007, pp. 1458–1463.

[98] F.-Y. Hsiao and C.-N. Lang, “Real-time target determination and tracking with an airborne video system,” in *Control Automation (ICCA), 11th IEEE International Conference on*, June 2014, pp. 1363–1368.

[99] F. Fonseca, M. Egenhofer, C. Davis, and K. Borges, "Ontologies and knowledge sharing in urban gis," *Computers, Environment and Urban Systems*, vol. 24, no. 3, pp. 251 – 272, 2000.

[100] A. Smirnov, T. Levashova, and N. Shilov, "Ontology-based knowledge sharing in flexible supply networks," *IFAC Proceedings Volumes*, vol. 41, no. 3, pp. 46 – 51, 2008.

[101] K. Munir and M. S. Anjum, "The use of ontologies for effective knowledge modelling and information retrieval," *Applied Computing and Informatics*, 2017.

[102] A. D. Iorio and D. Rossi, "Capturing and managing knowledge using social software and semantic web technologies," *Information Sciences*, pp. –, 2017.

[103] M. H. M. Noor, Z. Salcic, and K. I.-K. Wang, "Enhancing ontological reasoning with uncertainty handling for activity recognition," *Knowledge-Based Systems*, vol. 114, pp. 47 – 60, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705116303604

[104] C. Maio, G. Fenza, M. Gallo, V. Loia, and S. Senatore, "Formal and relational concept analysis for fuzzy-based automatic semantic annotation," *Applied Intelligence*, vol. 40, no. 1, pp. 154–177, Jan. 2014. [Online]. Available: http://dx.doi.org/10.1007/s10489-013-0451-7

[105] R. Navigli and P. Velardi, "From glossaries to ontologies: Extracting semantic structure from textual definitions," in *Proceedings of the 2008 Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge.* Amsterdam, The Netherlands, The Netherlands: IOS Press, 2008, pp. 71–87. [Online]. Available: http://dl.acm.org/citation.cfm?id=1563823.1563830

[106] R. Navigli, P. Velardi, A. Cucchiarelli, and F. Neri, "Quantitative and qualitative evaluation of the ontolearn ontology learning system," in *Proceedings of the 20th International Conference on Computational Linguistics*, ser. COLING '04. Stroudsburg, PA,

USA: Association for Computational Linguistics, 2004. [Online]. Available: http://dx.doi.org/10.3115/1220355.1220505

[107] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1535–1545. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145596

[108] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ser. ISWC'07/ASWC'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 722–735. [Online]. Available: http://dl.acm.org/citation.cfm?id=1785162.1785216

[109] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 1247–1250. [Online]. Available: http://doi.acm.org/10.1145/1376616.1376746

[110] R. Speer and C. Havasi, "Representing general relational knowledge in conceptnet 5," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23-25, 2012*, 2012, pp. 3679–3686.

[111] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 697–706.

[112] C. Matuszek, J. Cabral, M. Witbrock, and J. Deoliveira, "An introduction to the syntax and content of cyc," in *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, 2006, pp. 44–49.

[113] C. Kothari, J. Qualls, and D. Russomanno, “An ontology-based data fusion framework for profiling sensors,” in *Electro/Information Technology (EIT), 2012 IEEE International Conference on*, May 2012, pp. 1–6.

[114] J. Gomez-Romero, M. A. Patricio, J. Garcia, and J. M. Molina, “Context-based reasoning using ontologies to adapt visual tracking in surveillance,” in *Advanced Video and Signal Based Surveillance, 2009. AVSS ’09. Sixth IEEE International Conference on*, Sept 2009, pp. 226–231.

[115] M. Andersson, L. Patino, G. J. Burghouts, A. Flizikowski, M. Evans, D. Gustafsson, H. Petersson, K. Schutte, and J. Ferryman, “Activity recognition and localization on a truck parking lot,” in *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, Aug 2013, pp. 263–269.

[116] X. Zhang, C. Li, W. Hu, X. Tong, S. Maybank, and Y. Zhang, “Human pose estimation and tracking via parsing a tree structure based human model,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 5, pp. 580–592, May 2014.

[117] J. Garcia, A. Gardel, I. Bravo, J. L. Lazaro, and M. Martinez, “Tracking people motion based on extended condensation algorithm,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 3, pp. 606–618, May 2013.

[118] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V. K. Papastathis, and M. G. Strintzis, “Knowledge-assisted semantic video object detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1210–1224, Oct 2005.

[119] H. He and B. Upcroft, “Nonparametric semantic segmentation for 3d street scenes,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 3697–3703.

[120] Y. Li, Y. Guo, Y. Kao, and R. He, “Image piece learning for weakly supervised semantic segmentation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 4, pp. 648–659, April 2017.

[121] D. Cavaliere, S. Senatore, M. Vento, and V. Loia, "Towards semantic context-aware drones for aerial scenes understanding," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2016, pp. 115–121.

[122] D. Cavaliere, V. Loia, A. Saggese, S. Senatore, and M. Vento, "A human-like description of scene events for a proper uav-based video content analysis," *Knowledge-Based Systems*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705119301996

[123] D. Cavaliere, V. Loia, A. Saggese, S. Senatore, and M. Vento, "Semantically enhanced uavs to increase the aerial scene understanding," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 3, pp. 555–567, March 2019.

[124] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* Prentice Hall Professional Technical Reference, 2002.

[125] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

[126] M. R. Endsley and D. G. Jones, *Designing for situation awareness: An approach to user-centered design, Second Edition.* CRC press, 2012.

[127] D. Cavaliere, S. Senatore, and V. Loia, "A multi-perspective aerial monitoring system for scenario detection," in *2018 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, June 2018, pp. 1–6.

[128] W. Min, Y. Zhang, J. Li, and S. Xu, "Recognition of pedestrian activity based on dropped-object detection," *Signal Processing*, vol. 144, pp. 238 – 252, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168417303468

[129] L. Zhao, Y. Zhou, H. Lu, and H. Fujita, "Parallel computing method of deep belief networks and its application to traffic flow prediction," *Knowledge-Based Systems*, vol. 163, pp. 972 – 987,

2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705118305112

[130] G. D'Aniello, M. Gaeta, and T. P. Hong, "Effective quality-aware sensor data management," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 65–77, Feb 2018.

[131] G. Meditskos and I. Kompatsiaris, "iknow: Ontology-driven situational awareness for the recognition of activities of daily living," *Pervasive and Mobile Computing*, vol. 40, pp. 17 – 41, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S157411921630195X

[132] J. Bernad, C. Bobed, E. Mena, and S. Ilarri, "A formalization for semantic location granules," *International Journal of Geographical Information Science*, vol. 27, no. 6, pp. 1090–1108, 2013.

[133] G. Okeyo, L. Chen, and H. Wang, "Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes," *Future Generation Computer Systems*, vol. 39, pp. 29 – 43, 2014, special Issue on Ubiquitous Computing and Future Communication Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X14000399

[134] F. Zhang, D. Zhang, Y. Liu, and H. Lin, "Representing place locales using scene elements," *Computers, Environment and Urban Systems*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0198971517303903

[135] I.-H. Bae, "An ontology-based approach to adl recognition in smart homes," *Future Generation Computer Systems*, vol. 33, pp. 32 – 41, 2014, special Section on Applications of Intelligent Data and Knowledge Processing Technologies; Guest Editor: Dominik Ślęzak. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13000642

[136] A. Salguero and M. Espinilla, "Ontology-based feature generation to improve accuracy of activity recognition in smart environments," *Computers & Electrical Engineering*, vol. 68, pp. 1 – 13, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790617315483

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

[138] M. I. Ali, N. Ono, M. Kaysar, Z. U. Shamszaman, T.-L. Pham, F. Gao, K. Griffin, and 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*, vol. 42, pp. 19 – 37, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570826816300324

[139] N. Li, H. Guo, D. Xu, and 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)*, Oct 2014, pp. 2363–2367.

[140] D. Cavaliere, L. Greco, P. Ritrovato, and 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)*, Aug 2017, pp. 1–6.

[141] K. Avgerinakis, A. Briassouli, and Y. Kompatsiaris, "Activity detection using sequential statistical boundary detection (ssbd)," *Computer Vision and Image Understanding*, vol. 144, pp. 46 – 61, 2016, individual and Group Activities in Video Event Analysis. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314215002337

[142] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom sequence models for efficient action detection," in *CVPR 2011*, June 2011, pp. 3201–3208.

[143] P. Palmes, H. K. Pung, T. Gu, W. Xue, and S. Chen, "Object relevance weight pattern mining for activity recognition and segmentation," *Pervasive and Mobile Computing*, vol. 6, no. 1, pp. 43 – 57, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119209000996

[144] D. Cavaliere, S. Senatore, and V. Loia, “Proactive uavs for cognitive contextual awareness,” *IEEE Systems Journal*, pp. 1–12, 2018.

[145] L. Greco, P. Ritrovato, A. Saggese, and M. Vento, “Improving reliability of people tracking by adding semantic reasoning,” in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2016, pp. 194–199.

[146] A. Gangemi, “Ontology design patterns for semantic web content,” 11 2005, pp. 262–276.

[147] K. Gayathri, K. Easwarakumar, and S. Elias, “Probabilistic ontology based activity recognition in smart homes using markov logic network,” *Knowledge-Based Systems*, vol. 121, pp. 173 – 184, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705117300370

[148] A. Abdalla, Y. Hu, D. Carral, N. Li, and K. Janowicz, “An ontology design pattern for activity reasoning,” in *WOP*, 2014.

[149] M. M. Kokar, C. J. Matheus, and K. Baclawski, “Ontology-based situation awareness,” *Information Fusion*, vol. 10, no. 1, pp. 83 – 98, 2009, special Issue on High-level Information Fusion and Situation Awareness. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253507000218

[150] G. Meditskos, S. Dasiopoulou, V. Efstathiou, and I. Kompatsiaris, “Ontology patterns for complex activity modelling,” 07 2013.

[151] C. J. Matheus, M. M. Kokar, K. Baclawski, and J. Letkowski, “Constructing ruleml-based domain theories on top of owl ontologies,” in *RuleML*, 2003.

[152] G. Fu, “Fca based ontology development for data integration,” *Information Processing & Management*, vol. 52, no. 5, pp. 765 – 782, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030645731630019X

[153] A. Katib, V. Slavov, and P. Rao, “Riq: Fast processing of sparql queries on rdf quadruples,” *Journal of Web Semantics*, vol. 37-38,

pp. 90 – 111, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570826816000238

[154] R. H. Thomason, "Barwisejon. scenes and other situations. the journal of philosophy, vol. 78 (1981), pp. 369–397. barwisejon and perryjohn. situations and attitudes. the journal of philosophy, vol. 78 (1981), pp. 668–691. barwisejon and perryjohn. semantic innocence and uncompromising situations. the foundations of analytic philosophy, edited by frenchpeter a., uehlingtheodore e.jr., and wettsteinhoward k., midwest studies in philosophy, vol. 6, university of minnesota press, minneapolis1981, pp. 387–403." *The Journal of Symbolic Logic*, vol. 49, pp. 1403–1406, 12 1984.

[155] J. Barwise and J. Perry, *Situations and Attitudes*, 01 1983, vol. 78.

[156] J. Barwise, *The Situation in Logic*, ser. CSLI lecture notes. Cambridge University Press, 1989. [Online]. Available: https://books.google.it/books?id=aX7RKgvpJw8C

[157] V. Dragos and S. Gatepaille, "On-the-fly integration of soft and sensor data for enhanced situation assessment," *Procedia Computer Science*, vol. 112, no. Supplement C, pp. 1263 – 1272, 2017.

[158] M. Zeng, X. Wang, L. T. Nguyen, P. Wu, O. J. Mengshoel, and J. Zhang, "Adaptive activity recognition with dynamic heterogeneous sensor fusion," in *6th International Conference on Mobile Computing, Applications and Services*, Nov 2014, pp. 189–196.

[159] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 16–25, Sept 2006.

[160] M. Khan, S. Hassan, S. I. Ahmed, and J. Iqbal, "Stereovision-based real-time obstacle detection scheme for unmanned ground vehicle with steering wheel drive mechanism," in *2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, March 2017, pp. 380–385.

[161] M. R. Jabbarpour, H. Zarrabi, J. J. Jung, and P. Kim, "A green ant-based method for path planning of unmanned ground vehicles," *IEEE Access*, vol. 5, pp. 1820–1832, 2017.

[162] K. Ioannidis, G. Sirakoulis, and I. Andreadis, "Cellular ants: A method to create collision free trajectories for a cooperative robot team," *Robotics and Autonomous Systems*, vol. 59, no. 2, pp. 113 – 127, 2011.

[163] Y. Liu, M. Hoai, M. Shao, and T. K. Kim, "Latent bi-constraint svm for video-based object recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[164] X. Wang and Q. Ji, "Hierarchical context modeling for video event recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1770–1782, Sept 2017.

[165] M. Khan, K. Heurtefeux, A. Mohamed, K. A. Harras, and M. M. Hassan, "Mobile target coverage and tracking on drone-be-gone UAV cyber-physical testbed," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2017.

[166] N. Najva and K. E. Bijoy, "Sift and tensor based object detection and classification in videos using deep neural networks," *Procedia Computer Science*, vol. 93, no. Supplement C, pp. 351 – 358, 2016.

[167] D. Cavaliere, S. Senatore, and V. Loia, "Proactive uavs for cognitive contextual awareness," *IEEE Systems Journal*, pp. 1–12, 2018.

[168] J. Holsopple, M. Sudit, M. Nusinov, D. F. Liu, H. Du, and S. J. Yang, "Enhancing situation awareness via automated situation assessment," *IEEE Communications Magazine*, vol. 48, no. 3, pp. 146–152, March 2010.

[169] J. Lu, X. Yang, and G. Zhang, "Support vector machine-based multi-source multi-attribute information integration for situation assessment," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1333 – 1340, 2008.

[170] S. Klingelschmitt, F. Damerow, V. Willert, and J. Eggert, "Probabilistic situation assessment framework for multiple, interacting traffic participants in generic traffic scenes," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 1141–1148.

[171] M. Naderpour, J. Lu, and G. Zhang, “An abnormal situation modeling method to assist operators in safety-critical systems,” *Reliability Engineering & System Safety*, vol. 133, no. Supplement C, pp. 33 – 47, 2015.

[172] L. Snidaro, I. Visentini, and K. Bryan, “Fusing uncertain knowledge and evidence for maritime situational awareness via markov logic networks,” *Information Fusion*, vol. 21, no. Supplement C, pp. 159 – 172, 2015.

[173] I. O. Reyes, P. A. Beling, and B. M. Horowitz, “Adaptive multi-scale optimization: Concept and case study on simulated UAV surveillance operations,” *IEEE Systems Journal*, vol. 11, no. 4, pp. 1947–1958, Dec 2017.

[174] M. El-Zaher, F. Gechter, P. Gruer, and M. Hajjar, “A new linear platoon model based on reactive multi-agent systems,” in *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, Nov 2011, pp. 898–899.

[175] T. Iqbal, S. Rack, and L. D. Riek, “Movement coordination in human-robot teams: A dynamical systems approach,” *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 909–919, Aug 2016.

[176] A. R. da Silva and L. F. W. Goes, “Hearthbot: An autonomous agent based on fuzzy art adaptive neural networks for the digital collectible card game hearthstone,” *IEEE Transactions on Games*, vol. 10, no. 2, pp. 170–181, June 2018.

[177] B. Kosko, “Fuzzy cognitive maps,” *Int. J. Man-Mach. Stud.*, vol. 24, no. 1, pp. 65–75, Jan. 1986.

[178] ——, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.

[179] R. Taber, “Knowledge processing with fuzzy cognitive maps,” *Expert Systems with Applications*, vol. 2, no. 1, pp. 83 – 87, 1991.

[180] W. Pedrycz, “The design of cognitive maps: A study in synergy of granular computing and evolutionary optimization,” *Expert Systems with Applications*, vol. 37, no. 10, pp. 7288 – 7294, 2010.

[181] S. Lee and I. Han, "Fuzzy cognitive map for the design of edi controls," *Information & Management*, vol. 37, no. 1, pp. 37 – 50, 2000.

[182] J. A. Dickerson and B. Kosko, "Virtual worlds as fuzzy cognitive maps," in *Proceedings of IEEE Virtual Reality Annual International Symposium*, Sep 1993, pp. 471–477.

[183] C. E. Peláez and J. B. Bowles, "Using fuzzy cognitive maps as a system model for failure modes and effects analysis," *Information Sciences*, vol. 88, no. 1, pp. 177 – 199, 1996.

[184] C. D. Stylios and P. P. Groumpos, "The challenge of modelling supervisory systems using fuzzy cognitive maps," *Journal of Intelligent Manufacturing*, vol. 9, no. 4, pp. 339–345, Aug 1998.

[185] A. M. Khaleghi, D. Xu, Z. Wang, M. Li, A. Lobos, J. Liu, and Y.-J. Son, "A dddams-based planning and control framework for surveillance and crowd control via uavs and ugvs," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7168 – 7183, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417413005186

[186] J. A. Shaffer, E. Carrillo, and H. Xu, "Hierarchal application of receding horizon synthesis and dynamic allocation for uavs fighting fires," *IEEE Access*, vol. 6, pp. 78 868–78 880, 2018.

[187] E. Ertugrul, U. Kocaman, and O. K. Sahingoz, "Autonomous aerial navigation and mapping for security of smart buildings," in *2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)*, April 2018, pp. 168–172.

[188] L. Comba, A. Biglia, D. R. Aimonino, and P. Gay, "Unsupervised detection of vineyards by 3d point-cloud uav photogrammetry for precision agriculture," *Computers and Electronics in Agriculture*, vol. 155, pp. 84 – 95, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169917315491

[189] I. Pérez, F. Cabrerizo, S. Alonso, Y. Dong, F. Chiclana, and E. Herrera-Viedma, "On dynamic consensus processes in group decision making problems," *Information Sciences*, vol. 459, pp.

20 – 35, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025518303724

[190] J. Montero, "The impact of fuzziness in social choice paradoxes," *Soft Computing*, vol. 12, no. 2, pp. 177–182, Jan 2008. [Online]. Available: https://doi.org/10.1007/s00500-007-0188-5

[191] M. J. del Moral, F. Chiclana, J. M. Tapia, and E. Herrera-Viedma, "A comparative study on consensus measures in group decision making," *International Journal of Intelligent Systems*, vol. 33, no. 8, pp. 1624–1638, 2018.

[192] N. Capuano, F. Chiclana, H. Fujita, E. Herrera-Viedma, and V. Loia, "Fuzzy group decision making with incomplete information guided by social influence," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1704–1718, June 2018.

[193] G. Wei and M. Lu, "Pythagorean fuzzy power aggregation operators in multiple attribute decision making," *International Journal of Intelligent Systems*, vol. 33, pp. 169–186, 01 2018.

[194] Z. Zhen, D. Xing, and C. Gao, "Cooperative search-attack mission planning for multi-uav based on intelligent self-organized algorithm," *Aerospace Science and Technology*, vol. 76, pp. 402 – 411, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963817301736

[195] D. Zhang and H. Duan, "Social-class pigeon-inspired optimization and time stamp segmentation for multi-uav cooperative path planning," *Neurocomputing*, vol. 313, pp. 229 – 246, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231218307689

[196] P. Li and H. Duan, "A potential game approach to multiple uav cooperative search and surveillance," *Aerospace Science and Technology*, vol. 68, pp. 403 – 415, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963817309422

[197] F. Yan, K. Di, J. Jiang, Y. Jiang, and H. Fan, "Efficient decision-making for multiagent target searching and occupancy in an unknown environment," *Robotics and Autonomous Systems*,

vol. 114, pp. 41 – 56, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889018306882

[198] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25 – 33, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580517303825